
This work has been partially supported by the European Commission within the framework of the BIONETS project EU-IST-FET-SAC-FP6-027748, www.bionets.eu . This eBook constitutes project deliverable D0.2.3, and its diffusion level is marked as Public.

© BIONETS Consortium, 2010



Preface

On the 22nd of July 2003 the Future and Emerging Technologies Unit of the European Commission hosted a [brainstorming meeting on new communication paradigms for the year 2020](#). At that time, it seemed to some of us that the starting point in an advanced research programme in networking and communications was to view communications not just as links between individual nodes but, rather, as the ‘nervous system’ of the larger systems to which the nodes belong. Focussing on the behaviour of the overall system, whose dynamics are influenced by the characteristics of the underlying communication network, leads to a set of design and optimisation issues that are rather different to those conventionally considered in the design of computing/communications systems.

In particular, we started considering, as a good example to draw inspiration from, biological systems. In biological systems, typically, a plurality of communication means and media are present, which operate over multiple time scales, e.g. fast electrical pulses along the nerves, slower hormonal signals carried by the blood flow, and diffusion-based chemical signals between cells. The specialisation of function of different parts of biological systems is distributed spatially at different length scales, and each spatial scale tends to have its own characteristic speed of signal propagation. The various communication systems are not isolated from each other because ultimately they rely on the chemical and electrical properties of ions. The result of this integrated yet anisotropic, heterogeneous, and multi-scale architecture is the adaptive, complex behaviour often found in biological systems.

We soon recognised that architecting communications and computing systems *like* biological systems would require the introduction of radically different paradigms and design patterns. While the last decade has seen a flourishing of results and advances in understanding how biological systems work, turning such a knowledge into a purposeful design toolkit for computer scientists would require bridging a big gap. The first issue to be addressed was, in our view, the definition of suitable mathematical models, closely following the behaviour of biological systems but at the same time oriented towards the application to the design of networked, distributed ICT systems.

The flip-side of the mathematical point of view is the empirical approach. In this approach we explore new technologies in situ, meaning together with their users. Bringing the human users into the argument greatly amplifies the challenge of making sense of the already complex system described above. Suddenly we need to worry about what we are communicating and why, not just how: Is it information, multimedia content, voice? Is the link direct and point-to-point? Is there a feedback loop to the economic systems that support the development of the technology? Is there a feedback loop to the social, business, and/or e-government systems that influence the value metrics for services?

These considerations drove our work in the last few years. As a follow-up of the aforementioned brainstorming workshop, in late 2004 the European Commission launched a call for proposals on “Situating and Autonomic Communications”. Four projects were selected. The [BIONETS](#) project was one of the responses to the challenges outlined above, and ran for fifty months from January 2006 to February 2010. Research activities in BIONETS have been focussed around the definition of a biologically-inspired design toolkit for autonomic networks and services.

The central feature of the BIONETS approach – from an architectural perspective – was a ‘disconnected’ network, able to leverage short-range wireless communications and making extensive use of biologically-inspired techniques for handling control and management issues. The nature of the communication technologies addressed in BIONETS research makes it necessary to examine both the lower technology-centric levels as well as the upper user-centric levels. At the network level, the BIONETS system is composed of clouds of mobile nodes that are connected among themselves but potentially disconnected from any IP network or backbone, plus any number of sensors or embedded sources of contextual data. At the application and user level the BIONETS system is comprised of users interacting with each other and with a range of services and applications that are supported by the local cloud and by the sensors.

This eBook is a collection of some of the paradigms and foundations that have been identified as being relevant to the design, analysis and optimisation of autonomous wireless networks, in general, and to the architecture and methods envisaged for the project, in particular.

The term ‘paradigm’ has a slightly different meaning in different disciplinary contexts. Thomas Kuhn, a physicist, wrote about how the evolution of physical theories is strongly affected by the social context and the collaborative – or otherwise – dynamics of scientific endeavour.¹ His work has been very influential, especially in the social sciences and especially in the redefinition of the term ‘paradigm’, which can be paraphrased as ‘a body of theory combined with a community of practice and a set of research methodologies’. On the other hand, in many technical disciplines and in particular in computer science ‘paradigm’ is often used as a deeper and more general form of *model* or *theory*, without too much regard for the individuals engaged in its development or for the social processes its development might depend on. In this collection of articles we lean more towards the latter interpretation.

Our research addressed communication systems at three different levels: network protocol level, service level and user level. Solutions at different levels required the study of different paradigms. Whereas the lower two levels in the table below (protocols and services) can benefit from biological, physical, and mathematical models, the upper two levels (services and users) not only benefit from social science paradigms but actually *require* them to achieve a meaningful and sustainable integration of a BIONETS instance with its users. This is in line with current trends, which highlight the importance of the social dimension of technology in supporting sustainable business models and in enabling new value chains and value networks.

Level of BIONETS instance	Paradigms
User level	Social networks, new business models, eigenvector-based reputation metrics, community currencies, economics of sharing, network formation games
Service level	Artificial embryogeny, evolutionary algorithms, algebraic automata theory, category theory, logic, abstract algebra and group theory, security, distributed coordination
Protocol level	Machine learning, artificial chemistries, epidemic routing, evolutionary games, activator-inhibitor mechanisms, branching processes, free deconvolution and random matrix theory, road-traffic engineering, scale-free networks

From the point of view of the 2020 Vision of Communications that motivated the FET Proactive Initiative on “Situated and Autonomic Communications”,² the three most interesting aspects of the BIONETS architecture are:

- its ability to match closely the dynamic topology of co-located social groups (social and knowledge context);
- its ability to respond to local conditions through its reliance on sensors (physical and data context); and
- the long-term memory function that the sensors can support, which affords the system with something equivalent to a distributed, sub-symbolic intelligence.

Clearly the architectural aspect is only one part of the story. What is needed is also a flexible, dynamic, and adaptive service architecture that is able to keep up with the behaviour of the users and to meet their changing service needs. This is where BIONETS has invested a great deal of effort in a biologically-inspired approach that aims to endow the services and the underlying protocols with the ability to evolve.

¹ Kuhn, T (1996). *The Structure of Scientific Revolutions*, 3rd Ed, University of Chicago Press.

² Fabrizio Sestini, Thierry Van der Pyl: “Future and emerging technologies: a vision for tomorrow of EU IST research”. *Computer Communication Review* 35(2): 87-90 (2005).

Evolution, however, is only the tip of the iceberg: several additional paradigms from physics and biology are being examined and are summarised and contextualised in Parts 1 and 2 of this report in order to meet the challenging performance and adaptation requirements of the autonomic approach.

The purpose and context for this work is best understood by realising that the integration of the models examined and developed in this collection of chapters is not meant to produce a unified body of theory. By design, the research upon which this book is based is meant to look at different paradigms from a range of different disciplines (mainly biology, physics, mathematics, and social science). Some of the articles are specialistic, but most of them have been written in a tutorial style or as reviews, in an attempt to reach as wide and as interdisciplinary an audience as possible. Thus, these paradigms may find application in the design of distributed and autonomous computing and communication systems, far beyond the boundaries of the BIONETS project. We hope that this eBook will serve the ICT community at large, providing a source of references for unconventional design paradigms, able to meet the challenges arising from the Future Internet context.

E. Altman, P. Dini, D. Miorandi, D. Schreckling

Acknowledgements

The work reported in this eBook has been partially supported by the European Commission within the scope of the BIONETS Project EU-IST-FP6-027748 (www.bionets.eu). The authors acknowledge the support provided by the PO, Dr. G. Tselentis, and the project's evaluators (Prof. P. Van Mieghem, Prof. J. Timmis, Dr. M. Fehse, Dr. G. Aggelou, Prof. L. Cruickshank).

The editors and the authors further acknowledge all the partners in the BIONETS consortium, who contributed, in one way or another, to the successful completion of this eBook.

Table of Contents

Cover Page	I
Preface	II
Acknowledgments	IV
Table of Contents	V
Introduction	1

I Paradigms from Biology

Machine Learning for Intelligent Optimization <i>Mauro Brunato</i>	7
Evolutionary Computing and Artificial Embryogen <i>Lidia Yamamoto and Daniele Miorandi</i>	23
Evolutionary Games <i>Hamidou Tembine, Eitan Altman, Yezekael Hayel and Rachid El-Azouzi</i>	35
Activation-Inhibition Mechanisms for Distributed Coordination <i>Daniele Miorandi, Karina M. Gomez and Lidia Yamamoto</i>	44
Branching Processes and their Generalization Applied to Wireless Networking <i>Eitan Altman and Dieter Fiems</i>	54
On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security <i>Paolo Dini and Daniel Schreckling</i>	67
Algebraic and Categorical Framework for Interaction Computing and Symbiotic Security <i>Paolo Dini, Daniel Schreckling and Gábor Horváth</i>	105
Message Diffusion Protocols in Mobile Ad Hoc Networks <i>Ahmad Al Hanbali, Mouhamad Ibrahim, Vilmos Simon, Endre Varga and Iacopo Carreras</i>	177

II Paradigms from Physics

Free Deconvolution: from Theory to Practice <i>Florent Benaych-Georges and M�rouane Debbah</i>	201
Tools from Physics and Road-traffic Engineering for Dense Ad-hoc Networks <i>Eitan Altman, Pierre Bernhard, M�rouane Debbah and Alonso Silva</i>	225
Scale-Free Networks <i>Petri M�h�nen, Frank Oldewurtel and Janne Riihij�ervi</i>	241

III Paradigms from Social Science

Network Formation Games <i>Giovanni Neglia</i>	251
Eigenvector based Reputation Measures <i>Konstantin Avrachenkov, Danil Nemirovsky, Son Kim Pham, Roberto G. Cascella, Roberto Battiti and Mauro Brunato</i>	260
Historical-Interpretive Considerations about Money as a Unit of Value and Scale-Dependent Phenomenon <i>Silvia Elaluf-Calderwood</i>	279
Author Index	293
Subject Index	294

Introduction

The eBook contains three parts: paradigms from biology, from physics and from social science. To summarise the motivation of studying these paradigms, we cite Kelly [1]:

"There is currently considerable interest in the similarities between complex systems from diverse areas of physics, economics and biology, and it is clear that the study of topics such as noisy optimization and adaptive learning provide mathematical metaphors of value across many fields".

We describe below the various chapters, highlighting the specific contribution of the **BIONETS** project to the foundations and paradigms surveyed as well as to their applicability to the definition of novel techniques for designing autonomic networks and services.

Most paradigms in the book have been introduced to the context of networking and computing only recently. Some paradigms are novel ones and appear for the first time in a book. This is the case of, for example, tools from road traffic engineering. The eBook is the fruit of a combined effort by a large part of the participants of the European project BIONETS. Each chapter includes, in addition to the description of the paradigm, a discussion on its applicability to the networking/computing context.

1 Paradigms from Biology

Machine Learning and control under uncertainty. Stochastic search algorithms have been widely researched and used in the past decades for solving complex optimisation problems. The paradigm advocates the integration of machine learning techniques into stochastic search heuristics. In this chapter we provide the main motivations leading to this paradigm, followed by a survey on the application of machine learning to several types of stochastic search methods. The description of some recent work on this topic completes the chapter.

Evolutionary Computing and Artificial Embryogeny. In this chapter the authors present a review of state-of-the-art techniques for automated creation and evolution of software. The focus is on bio-inspired bottom-up approaches, in which complexity is grown from interactions between and among simpler units. First, the authors review Evolutionary Computing (EC) techniques, highlighting their potential application to the automated optimisation of computer programs in an online, dynamic environment. Then, they survey approaches inspired by embryology, in which artificial entities undergo a developmental process. The chapter concludes with a critical discussion and outlook for applications of the aforementioned techniques to the BIONETS environment.

Evolutionary games. Evolutionary games provide a theoretical framework to understand and predict the evolution of services, of protocols and of the architecture of decentralised networks whenever they evolve in a competitive context. By a competitive environment we mean that evolution occurs among several populations as a result of a process in which each population tries to improve its own fitness. Evolutionary games have been introduced by biologists to explain and predict evolution. This paradigm has the potential of further development when applied to engineering rules (or “eco-laws”) to achieve desired stability and efficiency objectives. In this chapter we first present the basic bricks of evolutionary games, and then provide various applications of the evolution of transport and of MAC layer protocols over wireless networks.

Activation-Inhibition Mechanisms for Distributed Coordination. The chapter deals with the use of activation–inhibition mechanisms for achieving coordinated behaviour in distributed communication systems. Mathematical models for activation–inhibition mechanisms are presented and analysed, and the possibility of reverse–engineering them for achieving given desired spatial patterns is discussed. A number of applications is reviewed, ranging from activation problems in wireless networks (access control, clustering) to distributed monitoring applications.

Branching Processes and their Generalization Applied to Wireless Networking. The chapter deals with the use of branching processes (and generalization thereof) to modelling problems in the wireless networking domains. Both discrete as well as continuous state space branching processes are considered,

and recent results obtained in the area (in particular with respect to correlated immigration) are presented. Closed-form results for the first two moments are presented. Application to ferry-based wireless local area networks, to epidemic message diffusion in delay-tolerant networks and to $G/PH/\infty$ queueing models are presented and discussed.

On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security. The material in this chapter is of an introductory nature, and is meant specifically to make some of the very abstract ideas of algebra and logic more accessible to researchers from other or more applied disciplines. Algebra and logic are very closely related. Because cell metabolic pathways can be mapped to automata whose algebraic properties can be analysed and classified, a bridge to translate the regularities in cell structure and behaviour into logic specifications of autonomic behaviour in services and protocols begins to appear possible. The main areas of application within BIONETS for this work is in the still-emerging concept of gene expression-based computing and security.

Algebraic and Categorical Framework for Interaction Computing and Symbiotic Security. This chapter builds on the previous one and continues the exploration and development of a theoretical and mathematical framework for biologically-inspired interaction computing, with security applications in mind. The chapter provides a broad conceptual discussion of the foundations and rationale for a theory of interaction computing and for the mathematical perspectives we advocate for its development. It then provides an exhaustive discussion of a permutation group example, in a tutorial style, to provide an intuitive basis for understanding the role of transformation semigroups as a basis of an algebraic theory of computation. The formalism of category theory is then presented in relation to the specification of automata behaviour and to its relationship to automata structure. The paper ends with a synthesis of the main insights gained to date in the emerging theory of interaction computing.

Message Diffusion Protocols in Mobile Ad Hoc Networks. The chapter surveys routing protocols in intermittently-connected mobile ad hoc networks. Due to low connectivity, direct paths may not exist between a source and a destination, and end-to-end communication has to rely on the mobility of terminals that relay packets. Tools from epidemiology have been used to analyse the process of spreading of copies of the packets in the network as well as mechanisms to stop it once the destination receives the packet.

2 Paradigms from Physics

Various paradigms that we describe in this survey allow one to obtain macroscopic properties of a system from the knowledge of the nature of microscopic interactions between basic elements of the system.

Free Deconvolution: From Theory to Practice. In many wireless communications applications, we encounter matrices whose entries are random. Examples in wireless communications include gain matrices appearing in MIMO channels which can be used for capacity calculations. In many cases, the distribution of the eigenvalues of the matrices converge to some non-random limit as the number of entries in the matrices grows. The theory of random matrices allows us to derive explicit expressions for those limits and thus provides a powerful tool for computing performance measures of systems with a large number of nodes. This approach is based in part on a recent theory called Free Probability which is surveyed as well.

Tools from Physics and Road-traffic Engineering for Dense Ad-hoc Networks. Spatial models from electrostatics and optics have been used for describing the paths followed by packets routed in dense ad-hoc network in a context in which mobiles are connected (in contrast with the approaches described in the chapter on “Message Diffusion Protocols in Mobile Ad Hoc Networks”, which are suitable for disconnected networks). The limit as the density of the network becomes large can therefore be described and its performance computed from simpler continuum models. We survey these approaches and propose alternative ones based on tools from road traffic engineering.

Scale-Free Networks. Scale-free distributions are known to appear often in various phenomena encountered in the natural world. From the BIONETS networking point of view, two cases of scale-free distribution have been identified and studied. The first one deals with the scale-free structure observed in organisational networks (usually expressed in terms of degree distribution of a graph representing the system’s relationships). This has been shown to arise in a variety of systems, from social networks to

the Internet and metabolic networks. The second one is concerned with the spatial distribution of node location in a network, which gives rise, in various cases, to fractal-like behaviour. The survey reviews methods and tools for characterising and analysing such cases, providing a useful model for (i) analysing various issues related to the BIONETS disappearing network (ii) providing a set of conditions for the occurrence of such structures, which are known to show peculiar properties (robust yet fragile).

3 Paradigms from Social Science

Network formation games. Network structure plays an important role in the performance of distributed systems, be it a group of friends, the World Wide Web or a business and commerce system. Researchers from various fields like physics, economics and social sciences have therefore been studying network formation. In the current Internet the network structure arises from interactions of agents at different levels. Internet Service Providers (ISPs) and different organisations decide autonomously which other entities in the Internet they want to be directly connected to. More recently Peer-to-Peer (P2P) networks and ad hoc networks have introduced new actors shaping the current Internet structure. All these agents (ISPs, peers,...) may have disjoint and competing interests, so game theory can provide useful mathematical tools to study the outcomes of their interactions. In particular there is a growing research trend on so-called network formation games, which explicitly consider players who can decide to connect to each other. In these games the network structure both influences the result of the economic interactions and is shaped by the decisions of the players. The purpose of this chapter is to provide the reader unfamiliar with this research area with the basic concepts, pointers to other surveys, and an overview of current results in the computer networks field.

Eigenvector based reputation measures. Reputation systems are indispensable for the operation of Internet mediated services, electronic markets, document ranking systems, P2P networks and ad hoc networks. Here we survey available distributed approaches to graph-based reputation measures. Graph-based reputation measures can be viewed as random walks on directed weighted graphs whose edges represent interactions among peers. We classify the distributed approaches to graph-based reputation measures into three categories. The first category is based on asynchronous methods. The second category is based on the aggregation/decomposition methods. And the third category is based on the personalisation methods which use local information.

Historical-Interpretive Considerations about Money as a Unit of Value and Scale-Dependent Phenomenon. This chapter reviews the main concepts and theories about money as a medium of exchange and economic models based on sharing of unused capital in the BIONETS environment. The rise of mobile technology and the multiple applications and services that can be provided directly to users offer the opportunity to leverage the economics of sharing and community currencies to create a wider user base where exchanges are economically motivated. Ubiquitous technology such as mobile devices raises new possible applications in a virtual world to understand, apply and negotiate these concepts. T-Node/U-Node separation, evolving applications, inherent user feedback, etc. promise to represent a wider technology platform providing more – and built-in – support for sharing and community currencies as the basis of new business models at the intersection between business services and social networking.

References

1. F. P. Kelly, "Network Routing", *Philosophical Transactions of the Royal Society*, A337, 343–367, 1991.

Part I

Paradigms from Biology

Machine Learning for Intelligent Optimization

Mauro Brunato

Information Engineering and Computer Science Department
University of Trento
I-38123 Trento, Italy
brunato@disi.unitn.it

Abstract. Stochastic optimization algorithms have been widely researched and used in the past decades for solving complex optimization problems. However, all such techniques require a lengthy phase of parameter tuning before being effective towards a particular problem; the tuning phase is performed by a researcher who modifies the algorithm's operating conditions according to his observations, therefore acting as a learning component. The Reactive Search paradigm aims at integrating sub-symbolic machine learning techniques into stochastic search heuristics, in order to automate the parameter tuning phase and make it an integral part of the algorithm execution, rather than a pre-processing phase.

The self-regulation property envisioned by the Reactive Search concept is motivated by the observation that in nature, and in particular in biological systems, feedback loops tend to be adaptive, i.e., they possess a learning component. In this chapter we provide the main ideas leading to the Reactive Search paradigm, followed by a survey on the application of Reactive Search concepts to several types of stochastic search methods.

1 Introduction

Optimization heuristics are motivated by the widespread belief that most interesting problems cannot be solved exactly within an acceptable time (e.g., time that is a low-degree polynomial function of the problem size). The last forty years have seen the introduction of many problem-specific methods: notable early examples are the Kernighan-Lin method for the graph partitioning problem [37] or the Steiglitz-Weiner heuristic for the Travelling Salesman Problem [58]. Ideas from such methods have been successfully extracted and applied to more general techniques, therefore called *meta-heuristics*, aimed at tackling problems in different domains by exploiting their similarity from a problem-solving viewpoint.

On the other hand, meta-heuristics suffer from the presence of operating parameters whose optimal values depend on the problem and on the particular instance being solved. All such techniques require therefore a (possibly long) phase of parameter tuning before being effective towards a particular problem, and in general the tuning phase is performed by a researcher who modifies the algorithm's operating conditions according to his observations, therefore acting as a learning component providing feedback to the algorithm itself. The Reactive Search paradigm advocates the integration of sub-symbolic machine learning techniques into stochastic search heuristics, in order to automate the parameter tuning phase, therefore making it an integral part of the algorithm execution, rather than an unaccounted preprocessing phase.

This chapter is organized as follows. Sec. 2 defines the application domain of optimization meta-heuristics. Sec. 3 provides a review of the basic meta-heuristic techniques. Sec. 4 introduces the framework of reactive search and presents a survey of recent developments. Applications of such techniques are explored in Sec. 5. Finally, Sec. 6 discusses the relevance of the proposed techniques within the BIONETS project.

2 Optimization problems

Throughout this chapter, a system will be described by a mathematical model whose parameter set (representing its degrees of freedom) is given by its *configuration space* \mathcal{X} ; an element $X \in \mathcal{X}$, representing a particular state of the system, is called a *configuration*. The criterion for measuring a configuration's fitness is defined within an *objective* function $f : \mathcal{X} \rightarrow \mathbb{R}$. In some cases, the criterion formulation naturally leads to a *penalty* measure, so that lower values of $f(X)$ correspond to a better configuration, sometimes

the function will represent a *fitness* measure. The aim of optimization is to find a configuration $X \in \mathcal{X}$ that minimizes $f(X)$ (if it represents a penalty measure) or maximizes it (if it represents fitness). A problem that can be expressed by the pair (\mathcal{X}, f) is called an *optimization problem*. If the configuration space \mathcal{X} is described in an implicit form by a system of equations and inequalities, then the problem is said to be *constrained*, otherwise it is said to be *unconstrained*.

This chapter considers *unconstrained* optimization problems where configurations can be mixed tuples of binary, integer or real values; the only form of constraints accepted in this context are the definition intervals of the scalar components, and \mathcal{X} can be expressed as a set product of intervals. While constrained problems are a superclass of unconstrained ones (the latter being the limit case with zero constraints), a constrained problem can always be transformed into an unconstrained one by accounting for constraint violations as penalties in the objective function. However, the constrained formulation can benefit from specific techniques, either analytical (e.g., Lagrange multipliers) or iterative (e.g., the simplex method and its many extensions), so that such reduction may decrease the chance of finding good solutions.

As a fundamental unconstrained example, let us consider the Maximum Satisfiability (MAXSAT) problem (see [25] for a classical survey), where n Boolean variables x_1, \dots, x_n are given. An *atom* is defined as either a variable x_i or the negation of a variable \bar{x}_i , and a (disjunctive) *clause* is the disjunction (logical OR) of a finite set of atoms. Given a finite set S of clauses, we ask what is the truth assignment to the n variables that maximizes the number of true clauses in S . In this case, the configuration space is $\mathcal{X} = \{0, 1\}^n$ (all possible truth assignments), while the objective function is

$$f(X) = \sum_{c \in S} \chi_c(X),$$

where $\chi_c(X)$ is 1 if the truth assignment X satisfies the clause c , 0 otherwise.

MAXSAT is still a very active research subject, and it is a useful benchmark for many techniques. Some are specifically aimed at characteristics of that domain; for instance, the *AdaptNovelty+* heuristic [59], although based on heuristics that span many applications, exploits peculiarities of the MAXSAT problem.

On the other hand, the last 40 years have seen the introduction of many heuristic methods that are applicable to different problems. We can define a *meta-heuristic* as a class of methods, operating on different domains, all sharing the same basic principle. Ideally, a meta-heuristic operates by receiving as input the pair (\mathcal{X}, f) describing the problem, and outputs an element $X \in \mathcal{X}$ selected in order to optimize f . Note that in this formulation the heuristic has no clue about the particular problem it is asked to solve; all such information has been used by the researcher to build the configuration space \mathcal{X} and the objective function f , which is treated as a “black box” by the algorithm.

Research work in this area was motivated by the observation that, when expressed in terms of fitness-function optimization, the properties of many problems can be described in terms of a common framework, which will be detailed in Sec. 2.1–2.2.

2.1 Neighbourhood structure

Very often, the problem that we are solving has a “canonical” topological structure defining when two configurations of the same instance may be considered “close” to each other. Considering such structure often carries advantages in the optimization process. Two configurations of a MAXSAT instance may differ by the value of a single variable. In this case, we naturally consider such configurations to be “similar,” and we expect most clauses (all clauses that do not contain the modified variable) to maintain the same truth value. The most common way of defining a topology on a configuration set is by defining a *neighbourhood* for every configuration. Consider the MAXSAT case: given a configuration $X \in \mathcal{X}$, let its neighbourhood be defined as all configurations that differ from it by at most one bit:

$$N(X) = \{X' \in \mathcal{X} | H(X, X') \leq 1\},$$

where $H(\cdot, \cdot)$ is the *Hamming distance* between two configurations, i.e., the number of bits by which they differ. A neighbourhood topology on \mathcal{X} is therefore the set

$$\mathcal{N} = \{(X, N(X)) | X \in \mathcal{X}\}.$$

An immediate advantage of considering this neighbourhood relationship between configurations is that, if the objective function is described by appropriate data structures, its value can be computed incrementally, saving CPU time.

A second, more important advantage of maintaining a neighbourhood relationship between configurations is that nearby solutions tend to have similar properties, so that if a configuration is “good” with respect to the objective function then its surroundings are worth exploring. In other words, once a topology \mathcal{N} is imposed upon a configuration space \mathcal{X} , the objective function can be expected to manifest some form of “regularity” with respect to it. In particular, the triplet $(\mathcal{X}, \mathcal{N}, f)$ can be analysed in terms of a *landscape* [35,60] in order to study the dynamics of solving algorithms on particular problems.

\mathcal{X}	(input)	Search space description
\mathcal{N}	(input)	Neighborhood structure
f	(input)	Objective function
t	(local)	Iteration counter
$X^{(t)}$	(local)	Current configuration at iteration t
\hat{X}	(local)	Best configuration found so far

1. **function** LocalSearch ($\mathcal{X}, \mathcal{N}, f$)
2. $t \leftarrow 0$
3. $X^{(0)} \leftarrow$ choose an element in \mathcal{X}
4. $\hat{X} \leftarrow X^{(0)}$
5. **while** some continuation condition is satisfied
6. $t \leftarrow t + 1$
7. $X^{(t)} \leftarrow$ choose an element in $N(X^{(t-1)})$
8. **if** $f(X^{(t)}) > f(\hat{X})$
9. $\hat{X} \leftarrow X^{(t)}$
10. **end if**
11. **end while**
12. **return** \hat{X}
13. **end function**

Fig. 1. The basic Local Search framework: the algorithm moves between neighbouring configurations

Looking in the proximity of known solutions is known as *intensification* of the search process (see [55] for an analysis) or *exploitation* of the solution. *Stochastic local search* techniques [31] aim at implementing this principle. The basic structure of a local search algorithm is shown in Fig. 1: the algorithm repeatedly moves from a configuration to a neighbouring one (the loop in lines 2.1–2.1), storing the best configuration found so far (lines 2.1–2.1). Heuristics differ in the choice of the initial configuration (line 2.1 of Fig. 1), of the subsequent neighbours (line 2.1), and the continuation condition (line 2.1).

2.2 Large-scale structure

In the previous section we have seen how a local search optimization algorithm can move between neighbouring configurations with the goal of improving the objective function by means of incremental changes.

Unfortunately, trying to improve the current solution by only performing incremental steps causes much of the problem’s search space to remain unexplored. Moreover, if the choice of the new configuration is always done towards the improvement of the objective function value, the system will finally get stuck in a locally optimal set of configurations. Note that optimization heuristics can be modelled as dynamic systems [29], describing their structure in terms of attractors (local optima) and attraction basins (sets of initial configurations leading to the same local optimum).

Once a local optimum has been achieved, further intensification of search in its neighbourhood becomes clearly counter-productive. Therefore, a *diversification*, or *exploration* strategy is needed. For ex-

ample, restarting from a new random configuration, in the hope to reach for a new portion of the search space, can be a good policy. However, in many practical cases the distribution of local optima should be taken into account: most real-world problems have in fact a rich structure (Chapter [Scale-Free Networks](#) provides a clear example of how neighbourhood relationships can generate rich global structures) that can be exploited by a more systematic search. Rather than restarting elsewhere, therefore, the algorithm should make an effort into trying to exit the current basin of attraction by executing local moves. In the context of local search meta-heuristics, this can be achieved in many ways, each leading to a broad class of heuristics.

Heuristics that do not base their behaviour on the local search paradigm (e.g., population-based or model-based algorithms) do not suffer from such local optimum entrapment; they will also be covered in [Sec. 3](#).

3 Basic optimization meta-heuristics

In this section we briefly summarize and review the basic groups of meta-heuristic algorithms for optimization. These heuristics differ both in the method chosen for intensification (some are based on local search, others implement mutation operators) and diversification.

3.1 Variable Neighbourhood Search

A problem formulation may allow for different neighbourhood definitions, each inducing a different topology on the search space. Moreover, there are cases when no optimal and fixed neighbourhood is defined for a problem because of lack of information. In many cases, however, adaptation of the neighbourhood to the local configuration is beneficial.

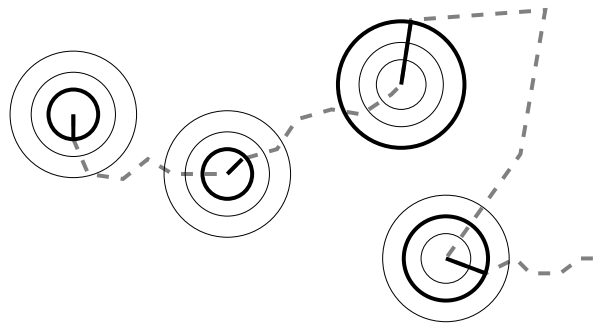


Fig. 2. Variable neighbourhood search: the used neighbourhood (bold circle around the current configuration) varies along the search trajectory.

The seminal idea of the Variable Neighbourhood Search (VNS) technique [24] is to consider a set of predefined neighbourhoods, and aim at using the most appropriate one during the search, as illustrated in [Fig. 2](#), where the possible neighbourhoods are represented as concentric circles around configurations, and the selected one is bold.

Many schemes for using the set of different neighbourhoods in an organized way are possible [27]. Variable Neighbourhood Descent (VND) uses a default neighbourhood first; other neighbourhoods are ranked in some order of importance and are used only if the default neighbourhood fails (i.e., the current point is a local minimum for it), and only until an improving move is identified, after which the algorithm reverts back to the default. If the ordering of the neighbourhoods is related to the strength of the perturbation, the algorithm will always use the minimum perturbation leading to an improvement. Variants of VND include REDUCED-VNS [44], a stochastic version where only one random neighbour is generated before deciding about moving or not, and SKEWED-VNS [26], where worsening moves are accepted if they lead the search trajectory sufficiently far from the current point. Other versions of VNS employ a stochastic move acceptance criterion, in the spirit of Simulated Annealing (see [Sec. 3.2](#)) as implemented

in the large-step Markov-chain version described in [41,40], where “kicks” of appropriate strength are used to exit from local minima.

3.2 Simulated annealing

The Simulated Annealing (SA) local search heuristic (see [38] for the seminal idea) introduces a *temperature* parameter T which determines the probability that worsening moves are accepted: a larger T implies that more worsening moves tend to be accepted, therefore diversification becomes larger. This behaviour is obtained by implementing variants of the following move acceptance rule, where $X^{(t)}$ is the configuration (solution) at iteration t and X' is a randomly chosen neighbour:

$$X^{(t+1)} \leftarrow \begin{cases} X' & \text{if } f(X') \leq f(X^{(t)}) \\ X' & \text{with probability } p = e^{-\frac{f(X')-f(X^{(t)})}{T}} \text{ if } f(X') > f(X^{(t)}) \\ X^{(t)} & \text{otherwise.} \end{cases} \quad (1)$$

Simulated Annealing has the properties of a Markov memoryless process: waiting long enough, every dependency on the initial configuration is lost, and the probability of finding a given configuration at a given state will be stationary and only dependent on the value of f . If T goes to zero the probability will peak only at the globally optimal configurations. This basic result raised high hopes of solving optimization problems through a simple and general-purpose method, starting from seminal work in physics [43] and in optimization [52,15,38,1].

Note that the dynamics of the Simulated Annealing heuristic depend on the value of T . If $T = 0$, only non-worsening paths are generated, leading to a local optima without any possibility of escape; if $T = \infty$, all moves are equally probable, leading to a random walk in the configuration space. The basic mechanism is a progressive reduction of T . The rate of reduction is called the *annealing* (or *cooling*) *schedule*, and many such schedules have been analysed for different problems [2,16,17].

3.3 Prohibition-based (Tabu) Search

The Tabu Search (TS) meta-heuristic [22] is based on the use of *prohibitions* as a complement to basic heuristic algorithms like local search, with the purpose of guiding the basic heuristic beyond local optimality.

Let us assume, for instance, that the configuration space is the set of binary strings with a given length n : $\mathcal{X} = \{0, 1\}^n$. Given the current configuration $X^{(t)}$, in Tabu Search only a subset $N_A(X^{(t)}) \subset N(X^{(t)})$ of neighbours is *allowed*, while the other neighbours are *prohibited*. The general way of generating the search trajectory that we consider is given by:

$$\begin{aligned} X^{(t+1)} &= \text{BESTNEIGHBOR}(N_A(X^{(t)})) \\ N_A(X^{(t+1)}) &= N(X^{(t+1)}) \cap \text{ALLOW}(X^{(0)}, \dots, X^{(t+1)}) \end{aligned}$$

The set-valued function ALLOW selects a subset of $N(X^{(t+1)})$ in a manner that in the general case depends on the entire past trajectory $X^{(0)}, \dots, X^{(t+1)}$. In practice, only a part of the search history is considered: for instance, a *prohibition period* T can be defined such that the ALLOW function only takes into consideration the T previous steps.

As a practical example, let us consider a problem whose configuration can be described by an n -bit binary string, such as MAXSAT. Given a configuration $X \in \mathcal{X}$, let its neighbours be all configurations at Hamming distance equal to one, i.e., all configurations obtained by flipping just one bit. A simple prohibition scheme is represented in Fig.fig:mlio:freezer: once a bit has been flipped, it cannot be flipped back to its previous value in the subsequent T iterations. In this case, the ALLOW function takes into account the bits that have been flipped in the last T iterations, and prohibits all neighbours that are

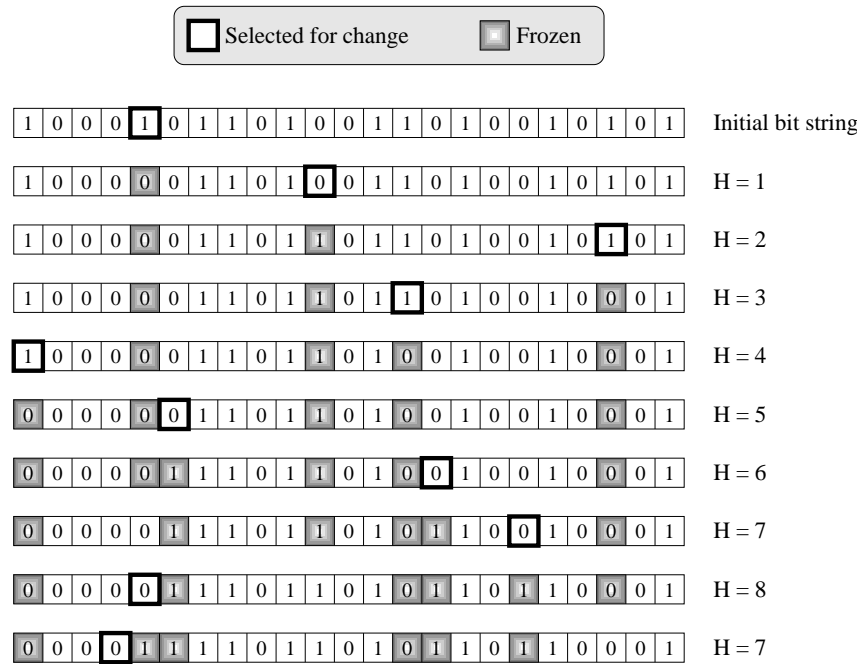


Fig. 3. Tabu Search: once a bit has been changed, it remains frozen for the subsequent T steps.

achieved by flipping any of those bits again. From a practical point of view, it is sufficient to record for every bit the last iteration it was flipped.

This simple technique is very effective at introducing a diversification dynamic into the search: in fact, it is apparent from Fig. 3 that, given the current configuration X , the T subsequent configurations will have a strictly increasing Hamming distance.

Some problems arising in TS that have been investigated are:

1. the determination of an appropriate prohibition period T for the different tasks,
2. the robustness of the technique for a wide range of different problems,
3. storing and using the past search history can be a computationally complex task.

In Section 4.3 we shall illustrate some possible solutions to these problems.

3.4 Genetic Algorithms

Among the many proposals about the adoption of natural, biological and evolutionary paradigms into Computer Science and simulation [4,54,21], Genetic Algorithms [56,30] try to introduce and adapt phenomena found in the natural framework of Evolution within the family of optimization techniques, placing themselves within the broader context of Evolutionary Computation (see also Chapter [Evolutionary Computing and Artificial Embryogeny](#)),

Evolutionary concepts are usually translated in the following way. An *individual* is a candidate solution and its genetic content (*genotype*) corresponds to its configuration $X \in \mathcal{X}$. A *population* is a set of individuals. The genotype of an individual is randomly changed by *mutations* (changes in X). The *suitability* of an individual with respect to the environment is described by the fitness function f . Finally, fitter individuals are allowed to produce a larger offspring (new candidate solutions), whose genetic material is a recombination of the genetic material of their parents.

As shown in Fig. 4, the basic GA maintains a population $\mathcal{P} \subseteq \mathcal{X}$ and makes it undergo a sequence of modifications produced by *operators*. The main types of operators are:

- **SELECT**: given a population $\mathcal{P} \subseteq \mathcal{X}$, produces a subset $\mathcal{P}' \subseteq \mathcal{P}$ according to the fitness of individuals and to stochastic factors. Usually, the value of the objective function for a given individual $X \in \mathcal{P}$ is used to determine the probability of its survival.

\mathcal{X}	(input) Search space description
f	(input) Objective function
\mathcal{P}	(local) Current population
\mathcal{P}'	(local) Temporary population under construction

```

1. function GeneticOptimization ( $\mathcal{X}, f$ )
2.    $\mathcal{P} \leftarrow$  choose elements from  $\mathcal{X}$ 
3.   while some continuation condition is satisfied
4.     Compute  $f(X)$  for each  $X \in \mathcal{P}$ 
5.      $\mathcal{P}' \leftarrow$  SELECT( $\mathcal{P}, f$ )
6.     for selected pairs  $X, Y \in \mathcal{P}'$ 
7.        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\text{COMBINE}(X, Y)\}$ 
8.     end for
9.     for random individuals  $X \in \mathcal{P}'$ 
10.       $X \leftarrow$  MUTATE( $X$ )
11.    end for
12.     $\mathcal{P} \leftarrow \mathcal{P}'$ 
13.  end while
14.  return best configuration ever visited
15. end function

```

Fig. 4. The basic Genetic Algorithm framework: the algorithm maintains a population of individuals that undergo selection, mutation and cross-combination. Standard bookkeeping operations such as optimum maintenance are not shown.

- **MUTATE:** given an individual $X \in \mathcal{X}$, produces a new individual X' by applying transformations consisting of point-wise changes, substring swappings or other mechanisms, depending both on the structure of the configuration space and the natural mechanism that is being emulated.
- **COMBINE:** given two individuals $X, Y \in \mathcal{X}$, a new individual Z is generated by combining parts of the genotype of X and Y . In the simplest case, every component of Z is chosen randomly from X or Y (uniform cross-over).

After the generation of an initial population \mathcal{P} (line 3.4 of Fig. 4), the algorithm iterates through a sequence of basic operations: the fitness of each individual is computed, then some individuals are chosen by a random selection process that favours elements with a high fitness function (line 3.4); a cross-over combination is applied to randomly selected survivors in order to combine their features (lines 3.4–3.4), then some individuals undergo a random mutation (lines 3.4–3.4). The algorithm is then repeated on the new population.

For example, in an optimization problem where the configuration is described by a binary string, such as MAXSAT, mutation may consist of randomly changing bit values with a fixed small probability and recombination of string X and Y may consist of building a new string Z so that

$$\forall i = 1, \dots, n \quad Z_i = \begin{cases} X_i & \text{with probability } 1/2 \\ Y_i & \text{with probability } 1/2. \end{cases}$$

Many hybridizations between Genetic Algorithms and Local Search algorithms have been proposed. The term *memetic algorithms* [45,39] has been introduced for models which combine the evolutionary adaptation of a population with individual learning within the lifetime of its members. The term derives from Dawkins' concept of a *meme* which is a unit of cultural evolution that can exhibit local refinement [18]. In these techniques, a team member can execute a more directed and determined exploitation of its initial genetic content (its initial position). This is effected by considering the initial genotype as a *starting point* and by initiating a run of local search from this initial point, for example scouting for a local optimum.

There are two ways in which such individual learning can be integrated: a first way consists of replacing the initial genotype with the better solution identified by local search (leading to a sort of *Lamarckian evolution*, where the parent transmits its own experience through its genes); a second way can be that of maintaining the original genotype, but modifying the individual's fitness by taking into account not the initial value but the final one obtained through local search. In other words, the fitness does not evaluate the initial state but the value of the “learning potential” of an individual, measured by the result obtained after the local search. These forms of combinations of learning and evolution are known as the *Baldwin effect* [28,61], and have the effect of changing the fitness landscape, while the resulting form of evolution is still Darwinian in nature.

3.5 Model-based heuristics

The main idea of model-based optimization is to create and maintain a *model* of the problem, whose aim is to provide some clues about the problem's solutions. If the problem is a function to be minimized, for instance, it is helpful to think of such model as a simplified version of the function itself, or a probability distribution defining the estimated likelihood of finding a good quality solution at a certain point. When used to optimize functions of continuous variables, model-based optimization is related to *surrogate optimization*, where a surrogate function is used to generate new sample points instead of the original function, which is in some cases very costly to compute, see for example [34], and also connected to the *kriging* [42] and *response surface* methodologies [47].

To solve a problem, the model is used to generate a candidate solution, which is in turn checked. The result of the check is used to refine the model, so that the future generation is biased towards better candidate solutions. Clearly, for a model to be useful it must provide as much information about the problem as possible, while being somehow “more tractable” (in a computational or analytical sense) than the problem itself. The initial model can be created through *a priori* knowledge or by uniformity assumptions.

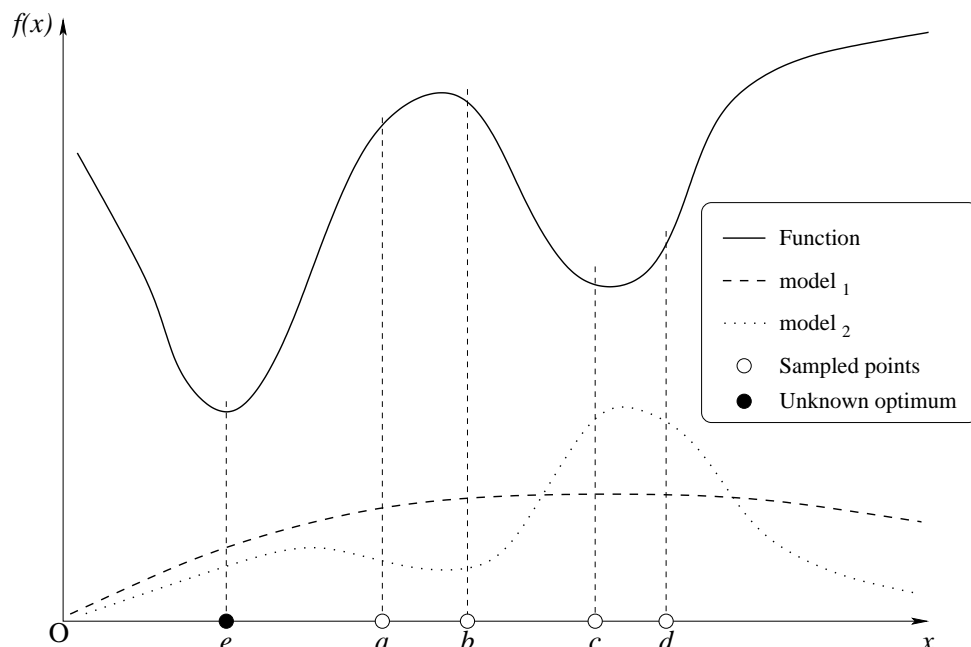


Fig. 5. Model-based search: one generates sample points from model₁ and updates the generative model to increase the probability for point with low cost values (see model₂). In pathological cases, optimal point e runs the risk of becoming more and more difficult to generate.

Although model-based techniques can be used in both discrete and continuous domains, the latter case better supports intuition. In Fig. 5 a function (continuous line) must be minimized. An initial model

(the dashed line) provides a prior probability distribution for the minimum (in case of no prior knowledge, a uniform distribution can be assumed). Based on this estimate, some candidate minima are generated (points a through d), and the corresponding function values are computed. The model is updated (dotted line) to take into account the latest findings: the global minimum is more likely to occur around c and d , rather than a and b . Further model-guided generations and tests shall improve the distribution: eventually the region around the global minimum e shall be discovered and a high probability density shall be assigned to its surroundings. The same example also highlights a possible drawback of naïf applications of the technique: assigning a high probability to the neighbourhood of c and d could lead to a negligible probability of selecting a point near e , so the global minimum would never be discovered. In other words, models tend to bias towards *intensification* of the search and must be corrected to ensure a significant probability of generating points also in unexplored regions.

Estimation of Distribution Algorithms (EDA) [46] have been proposed in the framework of evolutionary computation for modelling promising solutions in a probabilistic manner, so that the resulting model is used to produce the next generation of solutions. A survey in [51] considers population-based probabilistic search algorithms based on modelling promising solutions by estimating their probability distribution and using the model to guide the exploration of the search space.

f	(input)	Objective function
L	(input)	Number of bits in the configuration string
ρ	(input)	Learning rate
p	(local)	Generative model probability vector
\mathcal{P}	(local)	Current population
\mathcal{P}'	(local)	Selected fittest population
\hat{X}	(local)	Best configuration found so far

1. **function** PBIL (f, L, ρ)
2. $p \leftarrow \{0.5\}^L$
3. **while** some continuation condition is satisfied
4. $\mathcal{P} \leftarrow$ sample set generated with vector p
5. $\mathcal{P}' \leftarrow$ fittest solutions within \mathcal{P}
6. **for each** sample $X \in \mathcal{P}'$
7. $p \leftarrow (1 - \rho)p + \rho X$
8. **end for**
9. **end while**
10. **return** best configuration ever visited
11. **end function**

Fig. 6. Population-Based Incremental Learning: the algorithm updates a generative model by repeatedly generating a population of solutions and selecting the best individuals. Standard bookkeeping operations such as optimum maintenance are not shown.

A simple example of EDA-style model-based search is the Population-Based Incremental Learning (PBIL) algorithm [3], where individuals are described by a binary vector $\{0, 1\}^n$. The algorithm, shown in Fig. 6, maintains a probabilistic generative model with parameters $p = (p_1, \dots, p_n) \in [0, 1]^n$. This model is used to generate a population of candidate solutions \mathcal{P} (line 3.5). A selection of the fittest solutions (line 3.5) is used to modify the probability estimates of the solution's components, which are incrementally updated by a moving average (lines 3.5–3.5).

Another algorithm in the EDA framework is Mutual-Information-Maximizing Input Clustering (MIMIC) [19]. Given a function f to *minimize* within configuration space \mathcal{X} , the technique tries to set a convenient threshold θ and to estimate the distribution p^θ of solutions whose objective value is lower than θ . For selecting the proper threshold (notice that if θ is too low, no point is considered, while if it is too high all points are), the technique proposes a fixed percentile of a sampled subset. The algorithm then works

by progressively lowering the threshold, identifying the distribution of the fittest values. In order to compute the distribution within an acceptable CPU time, distributions are projected onto a reduced space by minimizing the Kullback-Leibler divergence.

4 Reactive search concepts

This Section contains a review of the main ideas that underlie the Reactive Search principle, on the basis of the common aspects listed in Sec. 2, and applied to the heuristics described in Sec. 3. More details can be found in [9,11].

The motivating observation for Reactive Search is the fact that all heuristics described up to now are *parametric*. It is difficult to assess, or even define, the *best* value for these parameters, which depends on the problem being addressed, on the particular instance, and on the aim of the researcher (do we want fast convergence towards the global optimum, or to an acceptable suboptimal solution?). The need for parameter tuning leads to a scenario where the (human) researcher adjusts the search algorithm's parameters in order to make it perform well on a given problem instance. Every time a new instance must be optimized, many optimization runs are needed in order to find the best parameter values.

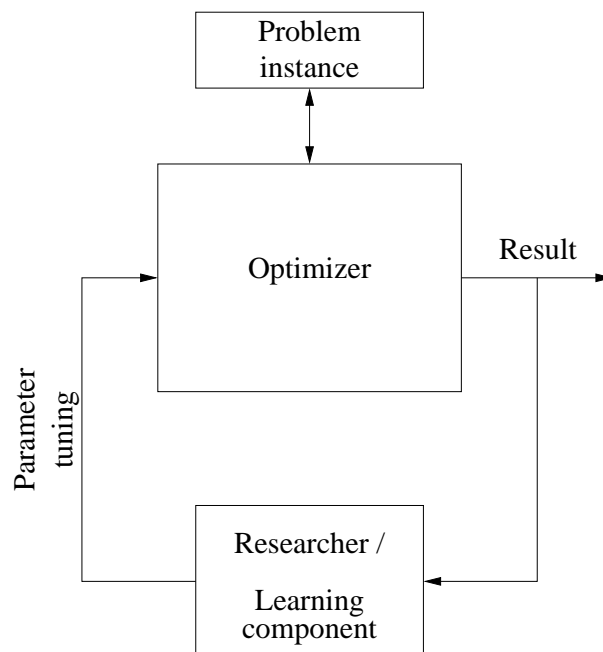


Fig. 7. Parameter tuning in optimization heuristics can depend on a human researcher; Reactive Search aims at machine learning-based automatic tuning.

The context is represented in Fig. 7, where the researcher acts as an intelligent feedback channel that is able to adjust parameters according to his observations about the algorithm's behaviour, his knowledge of previous runs, recognition of patterns in responses of the algorithm to parameter changes. In other words, he operates as a *learning component* of the algorithm.

Research on parameter-free optimization algorithms is therefore motivated by the need to exclude this tedious (and scientifically difficult to characterize) feedback phase operated by the researcher. Machine learning is a fundamental building block for such system.

The term *Reactive search* refers to the presence of this feedback component that allows the system to “react” to search events by modifying the algorithm's parameters during its execution, rather than in an off-line manner as in the human researcher's case, so that the learning component shown in Fig. 7 is an actual piece of software.

The machine learning component of an optimization algorithm can range, for instance, from a criterion to vary the balance of differentiation versus intensification according to the amount of explored

space, up to a heavy memory-based technique that stores all past history of the search in order to avoid re-exploring regions that were previously mined out.

The self-regulation property envisioned by the Reactive Search concept has been inspired by the observation that in nature, and in particular in biological systems, feedback loops tend to be adaptive, i.e., they possess a learning component which can be as simple as the variation of a chemical's concentration in a cell, up to the complexity of the brain cortex functionality in intentional reactions. In all cases, such adaptiveness can be characterized as "learning."

The advantage in automated parameter tuning is twofold: it provides complete documentation on the algorithm, which becomes self-contained and does not depend on external factors (i.e., human supervision), and removes the fine-tuning work from the researcher.

Reactive techniques have been proposed within many different optimization frameworks. Most reactive schemes are based on a common mechanism: first, the program maintains a history of the past search evolution. Next, this trace is used to identify local minimum entrapment situations (e.g., the same configurations are repeatedly visited), to identify patterns in the distribution of local minima, to build probabilistic models. Finally, the outcome of the learning process, which takes place along the execution of the search, is used to modify the basic search parameters which, in the stochastic search context, usually control the balance between intensification and diversification mechanisms.

In the following sections, applications of the Reactive Search concept to the basic optimization meta-heuristics are surveyed.

4.1 Variable Neighbourhood Search

An explicitly reactive VNS is considered in [12] for the Vehicle Routing problem with Time Windows (VRPTW), where a construction heuristic is combined with VND using first-improvement local search. The objective function used by the local search operators is modified to consider the waiting time to escape from a local minimum. A preliminary investigation about a self-adaptive neighbourhood ordering for VND is presented in [32]. Ranking of the different neighbourhoods depends on their observed benefits in the past and is dynamically changed during the search.

4.2 Simulated annealing

On-line learning strategies can be introduced in the algorithm's *cooling schedule*, by letting parameter T vary according to the search results. A very simple proposal [17] suggests resetting the temperature once and for all at a constant temperature high enough to escape local minima but also low enough to visit them. For example, at the temperature T_{found} when the best heuristic solution is found in a preliminary SA simulation. The basic design principle is related to: i) exploiting an attraction basin rapidly by decreasing the temperature so that the system can settle down close to the local minimizer, ii) *increase the temperature* to diversify the solution and visit other attraction basins, iii) decrease again after reaching a different basin. As usual, the temperature increase in this kind of non-monotonic cooling schedule has to be rapid enough to avoid falling back to the current local minimizer, but not too rapid to avoid a random-walk situation (where all random moves are accepted) which would not capitalize on the local structure of the problem.

Possibilities to increase the temperature to escape local optima include resetting the temperature to $T_{\text{reset}} = T_{\text{found}}$, the temperature value when the current best solution was found [50]. Geometric *re-heating* phases can be used [2], which multiply T by a heating factor γ larger than one at each iteration during a reheat phase. Enhanced versions involve a learning process to choose a proper value of the heating factor depending on the system state. In particular, γ is close to one at the beginning, while it increases if, after a fixed number of escape trials, the system is still trapped in the local minimum.

Modifications departing from the exponential acceptance rule (1) and adaptive stochastic local search methods for combinatorial optimization are considered in [48,49], where the authors note that adaptations should be done by the algorithm itself or by the user, by means of some learning mechanism. A simple adaptive technique suggested in [49] is the SEQUENCEHEURISTIC: a perturbation leading to a worsening solution is accepted if and only if a fixed number of trials could not find an improving perturbation

(this can be seen as deriving evidence of “entrapment” in a local minimum and activating reactively an escape mechanism). In this way the temperature parameter is eliminated. The positive performance of the SEQUENCEHEURISTIC in the area of design automation suggests that the success of SA is “due largely to its acceptance of bad perturbations to escape from local minima rather than to some mystical connection between combinatorial problems and the annealing of metals” [49].

“Cybernetic” optimization is proposed in [20] as a way to use probabilistic information for feedback during a run of SA. The idea is to consider more runs of SA running in parallel and to aim at intensifying the search by lowering the temperature parameter when there is evidence that the search is converging to the optimum value.

4.3 Tabu Search

In reactive versions of Tabu Search, most notably the Reactive Tabu Search (RTS) strategy [8], the prohibition period T is determined through feedback (i.e., *reactive*) mechanisms during the search. T is equal to one at the beginning (meaning that the inverse of a given move is prohibited only at the next step), it increases only when there is *evidence* that diversification is needed, it decreases when this evidence disappears. The evidence that diversification is needed is signalled by the repetition of previously visited configurations. All configurations found during the search can be stored in memory in a lazy learner fashion, or populate a more complex data structure. After a move is executed the algorithm checks whether the current configuration has already been found and it reacts accordingly (T increases if a configuration is repeated, T decreases if no repetitions occurred during a sufficiently long period).

RTS can be characterized as a memory-based search strategy where efficient data structures are needed to store and retrieve previously visited configurations in low amortized time. This can be obtained by combining hash tables with persistent data structures such as red-black trees [6,5].

4.4 Genetic and Population-based algorithms

In the Genetic Algorithms (GA) context, the concept of *metalevel Genetic Algorithm* [23] (meta-GA) has been proposed. Here the problem instance is solved by a population of GAs whose genotype is represented by their parameters. The fitness of these “individual” algorithms is measured by their performance in solving the problem, and an overlying GA is used to select the best individuals.

In this case, the overlying GA implements the feedback loop that is characteristic of the Reactive Search paradigm, see for example [57,62] for applications.

5 Biology-inspired applications

In this section we present two examples of reactive search applied to biology-inspired scenarios. In the first case (Section 5.1) we investigate a possible extension of a population-based approach by introducing “intelligent” individuals that apply a low-knowledge reactive scheme for local search. In the second part (Section 5.2) the distribution of local search algorithms on several machines is studied in terms of a trade-off between the rate of knowledge exchange (performed in an epidemic, peer-to-peer fashion) and the quality of the solution found.

5.1 A population-based approach to reactive search

A *clique* is a complete graph. The Maximum Clique problem asks for finding the largest complete subgraph embedded in a given graph. It is a well-known NP-hard problem: its decision version is NP-complete and non-approximable. Therefore, it is a well-studied testing ground for combinatorial metaheuristics, see for instance the DIMACS competition benchmark [33].

Following the recent introduction of Evolutionary Algorithms with Guided mutation for solving the maximum clique problem (EA/G [63]), the reactive and evolutionary algorithm R-EVO [10] has been proposed in the framework of estimation-of-distribution algorithms (EDA).

Classical model-guided mutation schemes use a model distribution to generate the offspring by combining the local information of solutions found so far with global statistical information (since they are based on a learning component, such algorithms, and in particular EA/G, can be considered within the reactive search paradigm). The R-EVO algorithm is placed in the same evolutionary framework, but considers more complex individuals, which modify tentative solutions by local search. In particular, the estimated distribution is used to periodically initialize the state of each individual based on the previous statistical knowledge extracted from the population. Each individual in R-EVO adopts a drastically simplified low-knowledge version of reactive local search (RLS) [7], with a simple internal diversification mechanism based on tabu-search, and a prohibition parameter proportional to the estimated best clique size. R-EVO is competitive with the more complex full-knowledge RLS-EVO which adopts the original RLS algorithm.

In [10] we study the combination of the EDA mechanism (which lies at the basis of the EA/G algorithm) with a stochastic local search algorithm derived from Reactive Tabu Search; we show that the resulting technique produces significantly better results than EA/G for most of the benchmark instances and is remarkably robust with respect to the setting of the algorithm parameters.

5.2 Gossiping Search Heuristics

While the use of distributed computing in search and optimization problems has a long research history, most efforts have been devoted to parallel implementations with strict synchronization requirements or to distributed architectures where a central server coordinates the work of clients by partitioning the search space or working as a status repository.

The distributed realization of a global optimization algorithm ranges from independent execution of instances (useful to mitigate very frequent heavy-tail behaviours) to complete synchronization and sharing of information (e.g., processes in a shared-memory multiprocessor machine). Between these two extremal cases, a wide spectrum of algorithms can be designed to perform individual searches with some form of loose coordination. Some paradigmatic cases of “collaborative search” are now collected under the BOINC initiative [53]. Although distributed, these projects are based on the repetition of a simple loop: every involved machine receives from a central server a subset of the search space (signal samples, number intervals), performs an exhaustive coverage of the subset, and reports the results, immediately receiving another search subset.

If the configuration space is to be searched by stochastic means, as opposed to the exhaustive search performed by the BOINC applications, centralized distribution methods are less appealing: having a central coordinator choose a partition of the search space may not be the best choice if the search space is large and only a small portion can be visited.

A completely distributed coordination scheme can be achieved in a peer-to-peer fashion using an epidemic protocol, where every node is aware of a small subset of peers and information is spread by exchange between nodes. In [13] the distributed implementation of global function optimization through decentralized processing in a peer-to-peer fashion is discussed. In our proposal, relevant information is exchanged among nodes by means of epidemic protocols. A key issue in such setting is the degradation of the quality of the solution due to the lack of complete information: different algorithm instances have a different snapshot of the global search status. A trade-off between message complexity and solution quality can be investigated. In particular, we concentrate on two algorithms for continuous global optimization: Particle Swarm Optimization [36] and the Memory-based Reactive Affine Shaker (M-RASH) [14].

In the Particle Swarm case, a set of searchers is guided by a very simple dynamics which depends on the position of the best value found by all searchers. Distributing this algorithm by implementing subsets of searchers in different machines involves the epidemic exchange of the global best coordinates. Fig. 8 shows some simulation results: PSO systems are distributed among a number of peer-to-peer nodes (5 to 30). Each node exchanges its local information with other randomly chosen nodes with a probability ranging from 10^{-4} to 1 at every iteration. While solution quality improves (i.e., minimum found approaches zero) with higher probability values, the system performs better than the equivalent algorithm on a single node (horizontal lines) even for fairly small exchange probability (less than 10^{-2}).

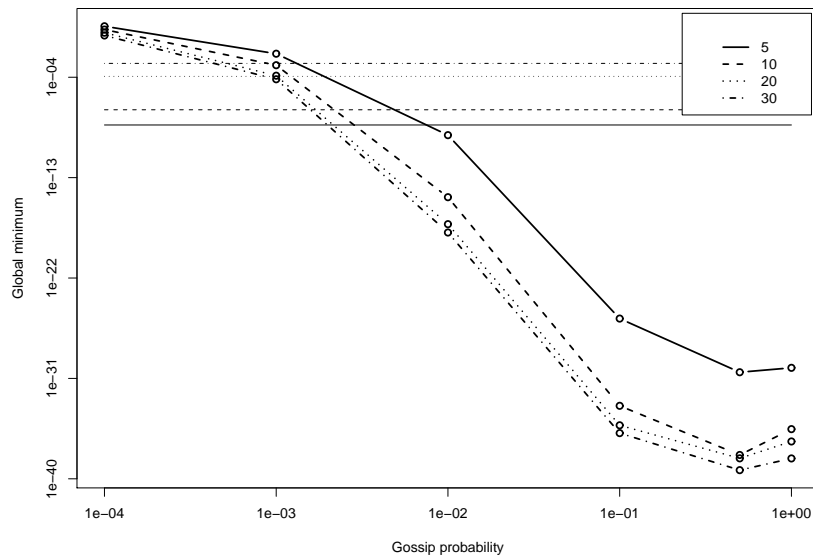


Fig. 8. PSO with various message exchange probabilities; horizontal lines refer to the single-node scenario

M-RASH implements a single searcher which maintains a model of the function being optimized. A form of loose coordination among M-RASH instances can be the exchange of information about the model, so that every searcher takes advantage of the knowledge gathered by others. The trade-off between the amount of information which can be effectively exchanged and the quality of the solution is being studied.

6 Relevance to the BIONETS context

Current research on reactive search is based on two nature-inspired paradigms, namely self-awareness and adaptation. While automated feedback loops are abundant in nature, we believe that the introduction of an explicit (although very mild) self-aware component is a viable complement to sophisticated optimization procedures. In particular, a machine learning component is inserted in the loop (see Fig. 7) to let the algorithm observe its own evolution in time and “react” appropriately by modifying its own behaviour according to such observation.

This is consistent to many feedback phenomena observed in Nature, and particularly in biological systems, where the feedback channel is often *adaptive*, in the sense that the same stimuli can correspond to different responses depending on some *memory* (maintained, e.g., by a chemical’s concentration or a neural configuration).

Self-awareness is also the inspiring paradigm of autonomic networks and systems, and optimization algorithms (in particular the distributed and population-based versions described in Section 5) can be particularly suited for application in such context.

7 Conclusion

The application of machine learning techniques to optimization algorithms is an active field of investigation, where many combinations of search schemes and machine learning techniques are possible. This chapter has provided motivation and examples of the application of machine learning schemes to optimization heuristics, with the purpose of automated parameter tuning. Some relevant applications oriented towards paradigms of interest by the BIONETS project have also been presented.

While the adoption of adaptive schemes based on machine learning techniques can improve the performance of an algorithm, an open issue in reactive search techniques is to establish a trade-off between the quality of the results and the computational burden of maintaining and searching the history-dependent data structures that lie at their foundations.

References

1. E.H.L. Aarts and J.H.M. Korst. Boltzmann machines for travelling salesman problems. *European Journal of Operational Research*, 39:79–95, 1989.
2. D. Abramson, H. Dang, and M. Krisnamoorthy. Simulated annealing cooling schedules for the school timetabling problem. *Asia-Pacific Journal of Operational Research*, 16:1–22, 1999.
3. S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. Technical report, School of Computer Science, Carnegie Mellon University, 1995. CMU-CS-95-141.
4. N.A. Barricelli. Numerical testing of evolution theories. *Acta Biotheoretica*, 16(1):69–98, 1962.
5. R. Battiti. Partially persistent dynamic sets for history-sensitive heuristics. Technical Report UTM-96-478, Dip. di Matematica, Univ. di Trento, 1996. Revised version, Presented at the Fifth DIMACS Challenge, Rutgers, NJ, 1996.
6. R. Battiti. Time- and space-efficient data structures for history-based heuristics. Technical Report UTM-96-478, Dip. di Matematica, Univ. di Trento, 1996.
7. R. Battiti and F. Mascia. Reactive local search for maximum clique: a new implementation. Technical Report DIT-07-018, University of Trento, 2007.
8. R. Battiti and G. Tecchioli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.
9. Roberto Battiti and Mauro Brunato. Reactive search: machine learning for memory-based heuristics. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, Computer and Information Science Series. Chapman & Hall / CRC, May 2007.
10. Roberto Battiti and Mauro Brunato. R-EVO: a reactive evolutionary algorithm for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, to be published, 2010.
11. Roberto Battiti, Mauro Brunato, and Franco Mascia. *Reactive Search and Intelligent Optimization*. Operations Research/Computer Science Interfaces Series, Vol. 45, Springer, November 2008. ISBN: 978-0-387-09623-0 Available at <http://www.reactive-search.org/thebook/>.
12. Olli Bräsly. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.
13. Mauro Brunato, Roberto Battiti, and Alberto Montresor. GOSH! gossiping optimization search heuristics. In *Learning and Intelligent Optimization Workshop LION 2007*, February 2007. Available at <http://dit.unitn.it/~brunato/publicazioni/gosh.pdf>.
14. Mauro Brunato, Roberto Battiti, and Srinivas Pasupuleti. A memory-based rash optimizer. In Ariel Felner, Robert Holte, and Hector Geffner, editors, *Proceedings of AAAI-06 workshop on Heuristic Search, Memory Based Heuristics and Their applications*, pages 45–51, Boston, Mass., 2006. ISBN 978-1-57735-290-7.
15. V. Cherny. A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–45, 1985.
16. T.-S. Chiang and Y. Chow. On the convergence rate of annealing processes. *SIAM Journal on Control and Optimization*, 26(6):1455–1470, 1988.
17. D.T. Connolly. An improved annealing scheme for the QAP. *European Journal of Operational Research*, 46(1):93–100, 1990.
18. R. Dawkins. *The selfish gene*. Oxford. Oxford University, 1976.
19. Jeremy S. de Bonet, Charles L. Isbell Jr., and Paul Viola. MIMIC: Finding optima by estimating probability densities. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 424. The MIT Press, 1997.
20. Mark A. Fleischer. Cybernetic optimization by simulated annealing: Accelerating convergence by parallel processing and probabilistic feedback control. *Journal of Heuristics*, 1(2):225–246, 1996.
21. A. Fraser and D.G. Burnell. *Computer models in genetics*. McGraw-Hill New York, 1970.
22. F. Glover. Tabu search - part I. *ORSA Journal on Computing*, 1(3):190–260, 1989.
23. JJ Grefenstette. Optimization of Control Parameters for Genetic Algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):122–128, 1986.
24. N. Mladenovic P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.
25. P. Hansen and B. Jaumard. Algorithms for the maximum satisfiability problem. *Computing*, 44:279–303, 1990.
26. P. Hansen, B. Jaumard, N. Mladenovic, and A. Parreira. Variable neighborhood search for weighted maximum satisfiability problem. *Les Cahiers du GERAD G-2000-62, Montreal, Canada*, 2000.
27. P. Hansen and N. Mladenovic. A tutorial on variable neighborhood search. Technical Report ISSN: 0711-2440, Les Cahiers du GERAD, Montreal, Canada, July 2003.
28. G.E. Hinton and S.J. Nowlan. How learning can guide evolution. *Complex Systems*, 1(1):495–502, 1987.
29. Tad Hogg. *Applications of Statistical Mechanics to Combinatorial Search Problems*, volume 2, pages 357–406. World Scientific, Singapore, 1995.
30. J.H. Holland. *Adaptation in Nature and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
31. H. H. Hoos and T. Stuetzle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

32. Bin Hu and G \ddot{A} $\frac{1}{4}$ nther R. Raidl. Variable neighborhood descent with self-adaptive neighborhood-ordering. In Carlos Cotta, Antonio J. Fernandez, and Jose E. Gallardo, editors, *Proceedings of the 7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics, malaga, Spain*, 2006.
33. David S. Johnson and Michael A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, volume 26. American Mathematical Society, 1996.
34. D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
35. S. A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.
36. J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. *IEEE Int. Conf. Neural Networks*, pages 1942–1948, 1995.
37. B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical J.*, 49:291–307, 1970.
38. S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
39. N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, Oct 2005.
40. Olivier Martin, Steve W. Otto, and Edward W. Felten. Large-step markov chains for the traveling salesman problem. *Complex Systems*, 5:3:299, 1991.
41. Olivier C. Martin and Steve W. Otto. Combining simulated annealing with local search heuristics. *ANNALS OF OPERATIONS RESEARCH*, 63:57–76, 1996.
42. Georges Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.
43. N. Metropolis, A. N. Rosenbluth, M. N. Rosenbluth, and A. H. Teller and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
44. N. Mladenovic. A variable neighborhood algorithm—a new metaheuristic for combinatorial optimization. *papers presented at Optimization Days, Montreal*, 112, 1995.
45. P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826, 1989.
46. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions i. binary parameters. In A. Eiben, T. Bäck, M. Shoenauer, and H. Schwefel, editors, *Parallel Problem Solving from Nature*, volume IV, page 178–187, 1996.
47. R.H. Myers and D.C. Montgomery. *Response surface methodology*. J. Wiley.
48. Surendra Nahar, Sartaj Sahni, and Eugene Shragowitz. Experiments with simulated annealing. In *DAC '85: Proceedings of the 22nd ACM/IEEE conference on Design automation*, pages 748–752, New York, NY, USA, 1985. ACM Press.
49. Surendra Nahar, Sartaj Sahni, and Eugene Shragowitz. Simulated annealing and combinatorial optimization. In *DAC '86: Proceedings of the 23rd ACM/IEEE conference on Design automation*, pages 293–299, Piscataway, NJ, USA, 1986. IEEE Press.
50. Ibrahim Hassan Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.*, 41(1-4):421–451, 1993.
51. M. Pelikan, D.E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.
52. Martin Pincus. A monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18(6):1225–1228, 1970.
53. The BOINC Project. <http://boinc.berkeley.edu/>.
54. I. Rechenberg. *Evolutionsstrategie*. Frommann-Holzboog, 1973.
55. Y. Rochat and E. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.
56. H.P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Inc. New York, NY, USA, 1981.
57. K. Shahookar and P. Mazumder. A genetic approach to standard cell placement using meta-geneticparameter optimization. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 9(5):500–511, 1990.
58. K. Steiglitz and P. Weiner. Algorithms for computer solution of the traveling salesman problem. In *Proceedings of the Sixth Allerton Conf. on Circuit and System Theory, Urbana, Illinois*, pages 814–821. IEEE, 1968.
59. Dave A. D. Tompkins and Holger H. Hoos. Novelty⁺ and adaptive novelty⁺. SAT 2004 Competition Booklet. (solver description).
60. E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.
61. D. Whitley, V.S. Gordon, and K. Mathias. Lamarckian Evolution, The Baldwin Effect and Function Optimization. In *Parallel Problem Solving from Nature—PPSN III: International Conference on Evolutionary Computation, Jerusalem, Israel*. Springer, 1994.
62. S.J. Wu and P.T. Chow. Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization. *Engineering Optimization*, 24(2):137–159, 1995.
63. Q. Zhang, J. Sun, and E. Tsang. An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, 9(2):192–200, 2005.

Evolutionary Computing and Artificial Embryogeny

Lidia Yamamoto¹, Daniele Miorandi²

¹ Computer Science Department, University of Basel
Bernoullistrasse 16, CH-4056 Basel, Switzerland

lidia.yamamoto@unibas.ch

² CREATE-NET

via alla Cascata 56/D

Povo, Trento – 38123, Italy

daniele.miorandi@create-net.org

Abstract. In this chapter we present a review of state-of-the-art techniques for automated creation and evolution of software. The focus is on bio-inspired bottom-up approaches, in which complexity grows from interactions among simpler units. First, we review Evolutionary Computing (EC) techniques, highlighting their potential application to the automated optimization of computer programs in an on-line, dynamic environment. Then, we survey approaches inspired by embryology, in which artificial entities undergo a developmental process. The chapter concludes with a critical discussion and outlook for applications of these techniques to the BIONETS environment.

1 Introduction

Building software that is able to continuously improve itself automatically is a common goal in artificial intelligence, software engineering, and other areas of computer science, including, more recently, autonomous systems and organic computing. The dream is to bring to computers the ability to constantly seek to learn and adapt, driven by a concrete purpose and motivation coming from the interaction with the real world [38].

Efforts in this direction follow a top-down or a bottom-up approach: Top-down approaches attempt to automate the reasoning process used in software engineering and design, from user requirements down to the code implementation. These include automatic program and protocol synthesis from specifications [27,36] and more recently, derivation of policy rules from high-level representations closer to natural language [51,47]. Bottom-up approaches look at how higher-level software functionality would emerge from lower-level interactions among simpler system units. Artificial Life (ALife), Evolutionary Computation, Swarm Intelligence and other areas focus on such bottom-up approach.

While the top-down approach seeks a formal model of software construction by humans, the bottom-up approach is essentially biologically-inspired. Even the most elementary life forms possess a level of robustness and adaptation far beyond current artificial systems, therefore it seems worthwhile to learn from biology in order to draw inspiration for the design of new systems.

In this chapter we provide a survey of bio-inspired approaches to such bottom-up creation of software functionality. Our focus is on dynamic, on-line processes where evolution and adaptation must happen continuously, during the operation of the system, as opposed to off-line, design-time optimization approaches. We investigate the potential of bio-inspired algorithms to obtain systems that are able to continuously pursue an optimum operation point without ever stopping. Such on-line optimization process involves the ability to self-organize into structures at multiple scales, analogous to cells, multicellular organisms, up to artificial ecosystems of interacting parts.

Numerous bio-inspired systems are available. A classification was proposed in [41], which positions them in a 3-D space defined by three axes, related to evolution of functionality, structural growth, and learning ability, respectively. We focus on the first two axes, represented mainly by evolutionary computation and developmental approaches related to embryology.

This chapter is organized as follows. In Sec. 2 we position our context within the classification adopted from [41]. In Sec. 3 we review the state-of-the-art in evolutionary computing with focus on on-line and dynamic environments. In Sec. 4 we present the two main research lines inspired by embryology: embryonics and artificial embryogenies. Sec. 5 presents a critical discussion on the possible combination of the aforementioned approaches. Sec. 6 concludes the chapter pointing out possible applications to BIONETS.

2 Context: The PO-Plane

A classification of bio-inspired systems was proposed in [41], positioning them in a 3-D space defined by three orthogonal axes. Although it was proposed ten years ago for hardware, its concepts remain valid today, and apply to software as well. We focus on two of the three axes, namely Phylogeny and Ontogeny. The third axis (Epigenesis), related to learning, covers techniques such as artificial neural networks and artificial immune systems, which are outside the scope of the present survey. The two remaining axes are defined as follows:

- *Phylogeny* or *phylogenesis* is the process of genetic evolution of species. Phylogenetic mechanisms are essentially non-deterministic, with mutation and recombination as major variation triggers. Artificial systems along this axis perform Artificial Evolution (AE) either in hardware (Evolvible Hardware) or in software (Evolutionary Computation). The latter will be described in Section 3.
- *Ontogeny* or *ontogenesis* is the process of growth and development of a multicellular organism from the fertilized egg to its mature form. Ontogeny is studied in developmental biology, which covers the genetic control mechanisms of cell growth, differentiation and morphogenesis. Artificial systems here go from simple replicators and self-reproducing systems to embryonics (mostly in hardware) and artificial embryogeny (mostly in software). These will be described in Section 4.

The POE classification is represented in Fig. 1, where some of the techniques which will be treated in this paper are positioned on the PO-Plane.

As predicted in [41], today combinations of both approaches, forming the so-called PO-Plane, are becoming more and more common. On one hand, an indirect encoding followed by a developmental process has shown to increase the scalability of evolutionary computing for complex problems. On the other hand, evolution enhances embryogenic systems with the potential of finding new solutions that were not preprogrammed. We conjecture that a combination of both is probably also essential to achieve the goal of on-line dynamic optimization, which is the focus of the present chapter: due to its highly non-deterministic nature, evolution alone would be too slow or insufficient to achieve this goal, while ontogenetic processes alone would lack the creation potential necessary to face new situations in a dynamic on-line environment. The potential of such combined PO-Plane approaches will be discussed in Section 5.

3 Evolutionary Computing

Evolutionary Computing, or *Evolutionary Computation* (EC) [17,15] derives optimization algorithms inspired by biological evolution principles such as genetics and natural selection. *Evolutionary Algorithms* (EAs) are meta-heuristics that can be applied to a variety of search and optimization problems. Existing EAs include: *Genetic Algorithms* (GAs), *Genetic Programming* (GP), *Evolutionary Programming* (EP) and *Evolution Strategies* (ES). They all model candidate solutions as a population of individuals with a genotype that is iteratively transformed, evaluated against a given fitness criterion, and selected according to the “survival of the fittest” principle, until an optimal solution is found. The difference among them lies in the way candidate solutions are represented, and on the search operators applied to obtain new solutions.

Recently, these existing iterative approaches are referred to as *Artificial Evolution* (AE) [4], in which biology concepts are applied in a very simplified way. In [4] the authors propose a new term *Computational Evolution* (CE) to reflect a new generation of bio-inspired computing [50] that builds upon new knowledge from biology and increased synergies between biologists and computer scientists.

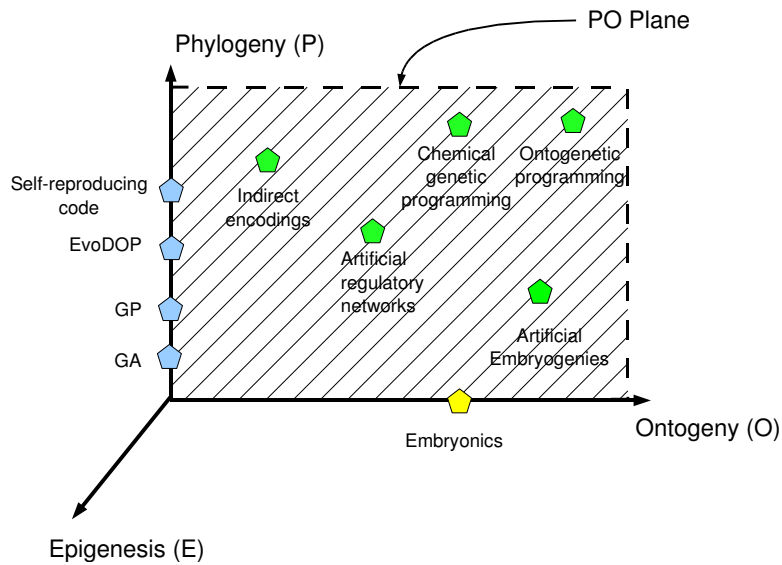


Fig. 1. The POE classification and some of the phylogenetic and ontogenetic approaches that will be treated in this paper.

AE is largely based on the so-called “Central Dogma of Artificial Evolution”, analogous to the “central dogma” of biology, in which information flows unidirectionally from DNA to proteins. This dogma is known today to be an over-simplification of reality. In CE, instead, one looks at the complex interactions that occur within the cell and beyond, such as genetic regulation and various other regulation mechanisms in the cell, the effects of interactions with the environment, symbiosis and competition in artificial ecosystems, and other highly dynamic processes which occur in many real-life problems. CE is of particular interest in on-line dynamic scenarios which are the focus of this survey. Note that there is no clear-cut border between AE and CE, but rather a gradual transition. For instance, the combination of phylogenetic and ontogenetic mechanisms positioned on the PO-Plane can be seen as a movement in the CE direction.

3.1 Genetic Algorithms

In a Genetic Algorithm (GA) [19] candidate solutions are represented as a population of individuals whose genotype is a string of solution elements (bits, characters, symbols, etc.). Strings typically have a fixed or bounded length, such that the size of the search space can be constrained. The goal of a GA is to find the optimum value of such string that optimises a given fitness criterion. An initial population of candidate strings is generated and evaluated against the fitness criterion. Multiple candidate solutions are then chosen (usually with a probability which depends on the respective fitness level) for giving rise to the next generation. This is accomplished by applying genetic operators (e.g., crossover and mutation) to such candidate solutions in order to create new variants.

3.2 Genetic Programming

Genetic Programming (GP) [21,6,24] applies the GA idea to evolve computer programs automatically. A GP algorithm is essentially the same as a GA, but the candidate solutions encode computer programs,

such that they can solve all instances of a problem, instead of optimizing for a particular instance as in GA.

GP typically evolves programs encoded in a linear (similar to assembly language) or tree representation (similar to functional languages such as LISP). Other representations are also possible, such as graphs [35,29], finite state machines [1,40,39], neural networks [30], and more recently, chemical programs [28,54].

When solving a problem by GP, one generally does not know the maximum size of the target solution program. Therefore, the genotype representation in GP generally allows for variable-length programs with unbounded size. The size of the search space in this case is infinite, so programs can in principle grow indefinitely. The *bloat* phenomenon was discovered early in the GP history, and refers to the fact that programs evolved by GP (especially tree-based GP) tend indeed to grow very large, with obvious shortcomings in terms of memory usage and execution efficiency. The bloat phenomenon is generally accompanied by an *intron growth* phenomenon, in which non-coding regions emerge, that have no effect on the program outcome. Although some authors pointed out that this phenomenon may also have positive effects, such as some protection against destructive crossover, it was mandatory to control code growth. Several methods were proposed for this purpose, such as parsimony pressure [26].

Recently, special attention has been devoted to indirect representations in which a genotype encoding is mapped onto a different phenotype representation. The goal is make GP solutions scale to complex problems without corresponding growth in program size. Indirect encodings may also provide additional robustness and evolvability, via redundant representations in which one phenotype may be expressed by more than one genotype, and via neutrality in representations, in which mutations in the genotype do not immediately affect the corresponding phenotype. This is especially important for on-line evolution. Indirect encodings will be briefly discussed in Section 3.6.

Most GP approaches can be placed along the “P” axis in the POE framework (Fig. 1). Some indirect encodings include a growth process which positions them on the PO-plane. An example is an Artificial Embryogeny, which will be discussed in Section 4.2.

3.3 Evolutionary Computing for Dynamic Optimization

EC techniques have been widely used for solving optimization problems in dynamic environments, in which the problem instance or the constraints may vary over time[20]. The aim is to introduce mechanisms able to “track” the optimal solution. This field is referred to as *Evolutionary Computing for Dynamic Optimization Problems* (EvoDOP). EC provides a natural framework for dynamic optimization, in that natural evolution is a continuous process. Most approaches in EC for dynamic optimization are based on the assumption that the changes in the problem settings are gradual, such that the previous population can be reused to search for the new optimum, without having to restart from scratch.

One of the main problems in EvoDOP is premature convergence: the population quickly converges to the optimum and tends to become very uniform, i.e. all solutions resemble the best one. In a static environment this is not an issue, since one can stop searching once a satisfactory solution is found. In a dynamic environment, premature convergence hinders the ability to search for new solutions. Proposed solutions for this problem include:

- Generate diversity after a change, e.g. through *Hypermutation*, i.e. artificially high mutation rates in response to a change.
- Maintain diversity throughout the run, e.g. through *random immigrants*, individuals that move between sub-populations.
- Implicit memory: usually takes the form of a redundant representation such as *diploid* or *polyploid* individuals with a dominance mechanism.
- Explicit memory: previously good solutions are stored in memory, and retrieved when the system encounters a previous situation for which the solution applied.
- Multi-population: Different sub-populations are spawned from a main population and assigned a subspace in which to track local optima. Several different promising subspaces can then be explored simultaneously.

- Anticipation and prediction: These are recent methods that attempt to predict the consequences of current decisions on the future of the system, such that informed decisions can be taken which will lead to improved results with high probability [8].

Although much has been done in EvoDOP in the GA domain, little has been explored in the GP domain. In [11] the authors show a multi-chromosome approach to GP based on Prolog programs. Multi-chromosomal GP is a polyploidy mechanism, thus a variant of implicit memory, which has been shown to achieve only mitigated results in EvoDOP. Indeed, the approach [11] is not applied to a dynamic environment. In nature, however, polyploidy mechanisms are extremely helpful, therefore it would be interesting to see how to improve the analogous artificial mechanisms to achieve an equivalent performance. Another research line would be to bring the most promising EvoDOP approaches to the GP domain, namely multi-population and anticipation. EvoDOP techniques are mainly situated along the “P” axis in the POE framework.

3.4 Self-Replicating and Self-Reproducing Code

Much attention has been paid in EC on self-replicating and self-reproducing code. In some cases, replication and reproduction have actually been considered synonymous, which they are not [41]. Replication is an ontogenetic, developmental process, involving no genetic operators, resulting in an exact duplicate of the parent organism. Reproduction, on the other hand, is a phylogenetic (evolutionary) process, involving genetic operators such as crossover and mutation, thereby giving rise to variety and ultimately to evolution.

The study of self-replicating software can be traced back to the pioneering work of John von Neumann in the late 40s on self-replicating automata. He set the basis for a mathematically rigorous study of self-replicating artificial machines based on cellular automata. Since then, several examples of self-replicating machines have been shown and elaborated [18]. Such machines are able to produce an identical copy of themselves, which means that the copy must also contain the part of the code that is able to produce a further copy of itself. Errors in the replication process are usually not allowed, and recovery from copy errors are thus in general not provided.

Self-reproducing code, on the other hand, involves a variation mechanism by which the new individual is not an exact copy of its parent. Self-reproduction thus requires some form of self-modification, which will be discussed below. Moreover it must include resilience to harmful replication errors in the form of a self-repair mechanism, or a selection mechanism able to detect and discard harmful code.

3.5 Self-Modifying Code

In a system that is required to constantly evolve and adapt, the ability to automatically modify or update its own code parts is essential. Since reliable and secure self-modification is still an open issue, self-modifying code has been banished from good practice software engineering.

However, self-modifying code plays a key role in EC and ALife, where evolution still occurs mostly in a simulated environment. In the case of EC, only the best programs which have been thoroughly tested via multiple fitness cases can be safely used. In the case of ALife, the main role of programs is just to survive, and since they remain in a virtual world there is no risk for the end user.

Evolvable instruction set virtual machines are used in most well-known ALife systems, such as Tierra and Avida. They resemble assembly language, which is easily self-modifiable: one can write on memory positions that include the own memory location of the code. This is used to evolve software that self-reproduces, adapts, seeks to survive, etc. A precursor of such machine language approach was Core Wars [13].

In the GP context, the *Push* family of programming languages [42] is designed for a stack-based virtual machine in which code can be pushed to a stack and therefore be manipulated as data. A variant of Push was used in *Autoconstructive Evolution* [43], where individuals take care of their own reproduction, and the reproduction mechanism itself can evolve (showing self-modification at the level of reproduction strategies). Recently [42], an enhancement of the language permitting the explicit manipulation of an

execution stack has been introduced. It has been shown to evolve iterative and recursive function which are non-trivial to be evolved in GP.

Ontogenetic Programming [45] is a developmental approach to GP in which the generated programs include self-modification instructions that enable them to change during the run. This is foreseen as an advantage for adaptation to the environment. To illustrate the concept, in [44] Ontogenetic Programming is applied to a virtual world game in which agents must find gold and survive multiple dangers and obstacles. It is shown that the ontogenetic version is able to evolve correct solutions to the game, where traditional GP fails to do so. This is an interesting example of hybrid approach located along the PO-Plane.

3.6 Indirect Encodings in Evolutionary Computing

It is well known in Evolutionary Computing that the representation of candidate solutions and the genetic operators applied to it play a key role in the performance of the evolutionary process. The genotype to phenotype mapping scheme is included in this representation problem, and it is well known in GP that indirect encodings like Cartesian GP [29] and Grammatical Evolution [31] can greatly help in obtaining viable individuals. Moreover they present a potential for encoding neutrality.

Neutrality occurs when small mutations in the genotype are likely not to affect the fitness of the corresponding individual. Such “silent” mutations, which modify the genotype while leaving the fitness unchanged, are called *neutral mutations*. Since the resulting changes are not subject to selection, their immediate impact is invisible. At first sight, they slow down evolution. However, over the long run, as neutral mutations accumulate, some genotypes may end up expressing a different solution with a potentially higher fitness. Neutrality provides a “smooth” way to explore the search space, and has been shown to potentially increase the evolvability of a population.

Indirect encodings may also be used to enhance the scalability of EC to complex problems: a compact genotype can express a large number of different phenotypes, such that the number of genes required to specify a phenotype may be orders of magnitude less than the number of structural units composing the phenotype. If coupled with developmental approaches (embryogeny, morphogenesis) it can encode phenotypes that grow from simple structures to more complex ones.

Many indirect encoding approaches include such a developmental process and can therefore be positioned on the PO-plane of the POE framework.

3.7 Approaches Based on Gene Expression

Many indirect encoding approaches have taken inspiration from gene expression in order to improve the performance of EC, especially GP. In these approaches, the process of decoding a genotype into a phenotype is analogous to expressing genes, and is controlled by a regulation or feedback mechanism.

Artificial Regulatory Networks have been shown to model the biological regulatory mechanisms in both natural [25] and artificial systems [23]. In [25] a genetic network exhibiting stochastic dynamics is evolved using a set-based encoding of systems of biochemical reactions. In [23] the regulatory network is represented with a genotype/phenotype binary encoding in which genes express proteins, which in turn control the expression of genes, unleashing large reaction networks that evolve by gene duplication and mutations. These networks are able to compute functions, such as a sigmoid and a decaying exponential.

Chemical Genetic Programming [34] proposes a feedback-based dynamic genotype to phenotype translation mechanism inspired by a cell’s dual step transcription-translation process from DNA to proteins. Using a chemical reaction mechanism, it dynamically builds the rewriting rules of a grammar used to translate a linear genotype into a tree phenotype. This leads to a highly dynamic and evolutive genotype to phenotype mapping: starting from a pool of simple grammar rules, the system evolves more complex ones and discards those that are not useful. While the concept itself seems promising, the encoding used and the algorithm itself are relatively complex, albeit applied to relatively simple problems.

Epigenetic Programming [48] associates a developmental process to GP, in which an Epigenetic Learning (EL) algorithm activates or silences certain parts of the genetic code. This is said to protect

individuals from destructive crossover by silencing certain genotypic combinations and explicitly activating them only when they lead to beneficial phenotypic traits. The authors show a 2-fold improvement in computational effort with respect to GP, on a predator-prey pursuit problem. Although in this approach a potentially large number of phenotypes can be expressed from a single genotype, this apparent increase in complexity is misleading, since all phenotypes are subsets of the original genotype.

Gene Expression Programming (GEP) [16] uses a linear genotype representation in the form of a chromosome with multiple genes. Each gene is translated into an expression tree, and trees are connected together by a linking function. Although inspired by gene expression, this approach does not include any dynamic feedback mechanism.

3.8 Chemical Computing Models and Evolution

Chemical computing models [3,10,33,14] express computations as chemical reactions that consume and produce computation objects (data or code). Objects are represented as elements in a *multiset*, an unordered set within which elements may occur more than one. The number of occurrences of a given element within the multiset is called the *multiplicity* of the element.

We believe that chemical models have a great potential for the class of on-line dynamic software optimization problems that we are aiming at. This is due to their inherent parallelism and multiset model, which permits several copies of instructions to be present simultaneously. We conjecture that a chemical language can express programs that can be more easily transformed and can become more robust to disruptions due to alternative execution paths enabled by a multiset model.

In this section we discuss some work in evolving programs using a chemical representation, which presents a new challenge for GP.

An *Algorithmic Chemistry* [5,52] is a reaction vessel in which instructions are executed in random order. In [5] the power of GP applied to an algorithmic chemistry on evolving solutions specific problems is shown. The authors point out the importance of the concentration of instructions, rather than their sequence. They start from a nearly unpredictable system in which execution of instructions at a random order leads to a random program output. This system is set to evolve by GP, including crossover and mutation of instructions placed in registers. After some generations, some order can be observed, and at the end of the evolutionary process a highly reproducible output is obtained, in spite of the random execution order.

The emergence of evolution in a chemical computing system is investigated in [28], using organization theory. Artificial biochemical signalling networks are evolved in [12] to compute several mathematical functions, using an evolutionary computation method based on asexual reproduction and mutations. Although the networks evolved in [12] show computational capacity, it does not seem trivial to extend their capabilities from mathematical functions to generic software actions.

We are working on our own chemical programming language [54,53] which we are extending with generic computation capacity and an evolution framework in which evolution will occur in an intrinsic and asynchronous manner as in nature. Variation operations such as mutation and recombination will occur within the individuals themselves as self-modifying code.

With such chemical systems it becomes possible to quantitatively regulate the behaviour of programs for evolution or adaptation purposes. An example of that is *Chorus* [2], a grammar-based GP system which uses a concentration table to keep track of concentrations of rules in the system. The rule with the highest concentration is picked for execution. The purpose is to obtain a system in which the absolute position of a gene (encoding a grammar rule number) does not matter. Such a system is then more resilient to genetic operators. In [53] we have proposed a code regulation system based on the control of the concentration of signals that activate or inhibit the expression of given genotypes according to their fitness. While [2] chooses the rule with the highest concentration, in [53] the choice is probabilistic: the chance of a variant being picked for execution is proportional to the concentration of its expression signals. While [53] is explicitly intended for on-line problems, to the best of our knowledge [2] has not been applied in this context.

4 Embryology

Embryology, in general, is a branch of developmental biology focusing on embryogeny, i.e., the process by which the embryo is formed and develops, from fertilization to mitotic divisions and cellular differentiation. The ability of embryos to generate complexity starting from a basic entity has generated a lot of attention in the computing field, since the ability to replicate *in silico* such process would enable researchers to break the complexity ceiling which limits the ability of conventional EC techniques.

The application of ideas from embryology (or, better: embryogenies) to artificial systems has been following two main research directions. One is *embryonics* (embryology plus electronics), an approach to improve fault tolerance in evolvable hardware by using a cellular architecture presenting dynamic self-repair and reproduction properties. Another one is *artificial embryogeny*, which aims at extending evolutionary computing with a developmental process inspired by embryo growth and cell differentiation, such that relatively simple genotypes with a compact representation may express a wide range of phenotypes or behaviours. These two directions reflect just different communities (hardware vs. software) rather than a clear conceptual partition, since the underlying concepts are common to both. Indeed, embryonics can be considered as a branch of artificial embryogeny which focuses on cell differentiation as an error handling mechanism in reconfigurable hardware, without necessarily covering evolutionary aspects.

4.1 Embryonics

The main goal of *embryonics* [9,37,32,49] is to embed extreme fault tolerance into electronic devices (e.g., FPGA arrays) while maintaining the redundancy (e.g., number of “spare” columns/rows in FPGA arrays) at acceptable levels. Approaches in this area have mostly focused on the use of *artificial stem cells*, i.e., cells which are able to differentiate into any specific kind of cell required for the organism to work. The approach is based on the flexibility offered by embryology-based mechanisms, in which there is no need to specify *a priori* the actions to be undertaken as a consequence of a fault detected. In FPGA arrays, specifying the reaction to each possible fault configuration would lead to poorly scalable designs, while at the same time resulting in a large overhead, due to the need of maintaining a large number of spare rows/columns. The systems devised in embryonics are based on the following two principles:

- Each cell (understood as the smallest indivisible building block of the system) contains the whole genome, i.e., has the complete set of rules necessary for the organism to work. Each cell is totipotent, i.e., can differentiate into any specific function and decides, based on the interaction with neighbouring cells, which functionalities (genes) need to be expressed.
- The system possesses self-organizing properties. Each cell monitors its neighbourhood and, upon detection of a faulty component, can return to the stem cell state and differentiate into another type of cell to repair the fault. (Some works have proposed to use solutions inspired by the mammalian immune system to implement this second functionality [9].) This step involves the availability of “spare” cells, which provide resources necessary to replace the faulty component.

The main difference between the embryonics approach to fault tolerance and classical approaches is that classical fault tolerance techniques tend to focus on simple replication as a way to achieve redundancy that can be used to recover from failures. In embryonics the information stored in neighbouring cells that might have differentiated to perform other functions may be used to recreate a lost functionality by re-expressing the corresponding genes that may be dormant in other cells. Such flexibility to switch functionality adds another level of robustness, as now not only cells with identical functionality can be used as backup or template to repair a failure, but also other cells with different functionalities can be used to recreate a lost one. In evolvable hardware, functionality is mainly expressed by the state of a cell (e.g. in the form of a configuration register), while in software it could also take the form of a computer program. Let us, for example, consider a distributed service, i.e., a service whose outcome comes from the interaction of different components running on different machines. Distributed services are prone to errors related to the possible faults of one (or more) of the machines where the components reside and

run. This is particularly important in open uncontrolled environments, where the resources used for providing the service do not reside on dedicated servers but are the spare resources possibly present in user's desktop or even mobile devices. A reactive and efficient self-repair or self-healing ability is essential in this context. It is not enough to rely purely on classical fault tolerance, where failure modes must be pre-engineered into the system. Neither can we rely exclusively on evolutionary mechanisms, which tend to be slow and unreliable, requiring a resilience mechanism of their own. Clearly, embryonics provides a middle ground in which diversity can be exploited for quick repair and re-adaptation, without changing the underlying (potentially evolvable) genotype. This will involve the construction of artificial stem cells, in the form of representation of the *complete* service instructions to be used. Such artificial stem cells shall be spread in the network, where they shall differentiate (for example following a reaction-diffusion pattern) into the different components needed for performing the service. Upon detection of a fault, they could re-enter the embryo state, for differentiating again into the required functionalities, expressing the necessary genes.

The Embryonics approach does not encompass any evolutionary aspect. Therefore, in terms of the classification introduced in Sec. 2, we can position it as a pure ontogenetic approach.

4.2 Artificial Embryogeny

Artificial Embryogeny [46] is a branch of Evolutionary Computing (EC) in which compact genotypes are expressed into phenotypes that go through a developmental phase that may cause them to differentiate to perform specific functions. Indeed, researchers have recognized that “conventional” EC techniques (like GA, GP, Evolutionary Strategies, etc.) present scalability problems when dealing with problems of relevant complexity [17]. The issue is that the size of the genotype representing possible solutions in the search space turns out to grow fast as the complexity of the organism/behaviour to be optimized grows. One solution studied in such approach has been to add one more level of abstraction. In such case, the genotype does not code the solution itself, but it codes recipes for building solutions (i.e., phenotypes). In this way, a genotype change does not imply a direct change in the solution, but in the way solutions are decoded from the genotype and further grown from an initial “seed” (the embryo).

In the case of GP, such indirect genotype encodings play an important role in obtaining viable individuals (i.e., syntactically correct programs suitable to be executed) via genetic transformations such as crossover and mutation. Approaches in the GP area are classified according to the genotype representation and decoding: grammatical evolution and developmental/ontogenetic GP [17]. In the first case, the genotype is a grammar that comprises a set of rewriting rules which are applied until a complete phenotype is obtained [31]. Since grammar production rules are applied to obtain the program, the derived program is syntactically correct by construction. In the second case, the genotype contains a set of instructions/transformations which are applied repeatedly on an embryonic entity to obtain a full organism. One of the most prominent examples of the second case is Ontogenetic Programming [45] (see Section 3.5), which produces self-modifying programs that are highly adaptive.

Note that although grammatical evolution is an indirect encoding approach, it is not performing embryogeny per se, as the generated individuals do not necessarily continue to develop after the phenotype is expressed. One could easily imagine a grammar to express self-modifying programs (for instance, a grammar that encodes for the stack-based linear programs in [45,44]) such that grammatical evolution then becomes part of the full cycle of evolution, gene expression, development and adaptation. However this is orthogonal to the developmental process implied in artificial embryogeny.

Other grammar approaches outside the GP context have an inherent growth model which has been associated with artificial embryogeny. Chapter 2.1 of [46] presents a survey of these grammatical approaches to artificial embryogeny. A simple one is to use L-Systems. L-Systems, or Lindenmayer Systems, express fractal-like objects using a grammar where the production rules may or may not contain parameters that determine how the structure will grow. Why is this form of grammar closer to embryogeny than grammatical evolution? Since L-Systems encode structures as opposed to executable programs, any intermediate step in the expansion of an L-System is a valid structure (thus a valid individual in growth process), while in grammatical evolution the first valid program that can be executed is one in which

all production symbols (non-terminals) have been rewritten into terminal ones. On the other hand, L-Systems have not been designed with evolution in mind, although they have been later used for this purpose.

Studies on artificial embryogeny [7] have reported a clear advantage of an indirect encoding over a direct one such as tree-based GP. On the other hand, further experiments [22] report that the indirect approach actually takes much longer to run, and show cases where tree-based GP outperforms the indirect approach and vice-versa. What remains consistent across different experiments is that in general, indirect encodings perform best when the problem is complex. There is therefore a trade-off between the computational resources needed for performing embryogeny and the complexity of the problem to be tackled, which needs to be carefully accounted for, especially in the presence of resource-constrained devices.

Artificial embryogenies may encompass both an ontogenetic as well as a phylogenetic aspect: as such, they lie in the PO-plane.

5 Discussion: EC and Embryology: Common Synergies

While Evolutionary Computing and Embryology-based approaches pertain both —broadly speaking— to the same research area, and while there is a considerable overlap in the research communities involved (with the notable exception of embryonics), there are many synergies which have not been exploited so far. Indeed, embryology-based approaches are still in their infancy, having been mostly applied to solve simple problems or toy-cases, and could profit from the 30+-years experience and insight gained from research in EC. At the same time, it is worth remarking that a thorough understanding of all the combinations possible on the PO-Plane is still missing. Nonetheless, embryology-based approaches have the potential to complement genetic-based EC techniques by providing a different level of system dynamics. While, indeed, one of the advantages of artificial embryogenies is related to the possibility of encoding complex behaviours in a parsimonious way (thus tackling the scalability problems encountered by standard GA/GP techniques when applied to many real-world problems), it seems to us that another advantage would be the possibility of having a much faster system dynamics, obtained through a fast growth process. This is extremely important in applications requiring near real-time adaptability, a feature badly supported by pure evolutionary approaches. Such an issue is extremely important in the perspective of the BIONETS project, where services are expected to be able to self-organize and adapt to varying working conditions in an extremely timely manner.

Further, embryology-inspired approaches can sustain interactions with the environment in a natural way (embedding a self repair mechanism in most cases, see e.g. embryonics, and a form of adaptation in some others), therefore complementing the natural selection process at the hearth of EC techniques. This is particularly important when EC techniques are applied in an on-line fashion. Indeed, EC techniques can easily lead to the creation, in the evolutionary paths, of organisms which are not able to perform the expected operations. In conventional EC applications this is not an issue, as intermediate solutions are not turned into a working system but only the final result of the evolutionary process is used. On the contrary, in an on-line evolution also intermediate solutions are used for performing the system's operations. The ability to repair automatically and/or to sustain the presence of faulty components becomes therefore a critical one for ensuring purposeful system operations.

We believe that a combination of the two approaches can be used to effectively design distributed, autonomic software systems as addressed within the BIONETS project.

6 Conclusion

In this chapter, we have presented a survey of existing approaches in Evolutionary Computing and Embryology-inspired techniques. While these techniques have been proven useful in a variety of problems, we are still far from the application of such techniques for creating and evolving software in an on-line and dynamic way.

In the perspective of the BIONETS project, there are therefore issues of fundamental nature which need to be tackled before moving to the application of such techniques for solving the problems related to

autonomic computing and communication systems. Nonetheless, there is a considerable body of works in the area which can provide insight into the design of novel, bio-inspired techniques which may prove to overcome conventional static solutions by enriching them with the possibility of evolving - in an unsupervised manner - new configurations, providing enhanced flexibility, robustness and resilience.

At the same time, CE and the forthcoming second generation of bio-inspired systems [50] (which is looking more closely into biology in order to draw inspiration from more accurate biological models as opposed to the initial coarse-grained, highly simplified ones at the basis of EC) brings with it the promises of moving one step further in the creation of suitable computational models for self-creating software code. While this may take long to come, it is our belief that the area of bio-inspired solutions to autonomic computing represents the most promising and viable approach to tackle such problems.

References

1. S. G. Araújo, A. C. P. Pedroza, and A. C. Mesquita. "Evolutionary Synthesis of Communication Protocols". 10th International Conference on Telecommunications (ICT 2003), 2:986–993, February-March 2003.
2. R. M. A. Azad. "A Position Independent Representation for Evolutionary Automatic Programming Algorithms - The Chorus System". PhD dissertation, University of Limerick, 2003.
3. J.-P. Banâtre, P. Fradet, and Y. Radenac. "A Generalized Higher-Order Chemical Computation Model with Infinite and Hybrid Multisets". In 1st International Workshop on New Developments in Computational Models (DCM'05), pages 5–14, 2005. To appear in ENTCS (Elsevier).
4. W. Banzhaf, G. Beslon, S. Christensen, J. Foster, F. Képès, V. Lefort, J. Miller, M. Radman, and J. Ramsden. "From Artificial Evolution to Computational Evolution: A Research Agenda". Nature Reviews Genetics, pages 729 – 735, 2006.
5. W. Banzhaf and C. Lasarczyk. "Genetic Programming of an Algorithmic Chemistry". In Genetic Programming Theory and Practice II, O'Reilly et al. (Eds.), volume 8, chapter 11, pages 175–190. Kluwer/Springer, 2004.
6. W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. "Genetic Programming, An Introduction". ISBN 155860510X. Morgan Kaufmann Publishers, Inc., 1998.
7. P. J. Bentley and S. Kumar. "Three Ways to Grow Designs: A Comparison of Embryogenies for an Evolutionary Design Problem". In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), pages 35–43, Orlando, Florida USA, 1999.
8. P. A. N. Bosman. "Learning, anticipation and time-deception in evolutionary online dynamic optimization". In Proc. Genetic and Evolutionary Computation Conference (GECCO 2005), pages 39–47, Washington DC, USA, June 2005.
9. D. Bradley, C. Ortega-Sanchez, and A. Tyrrell. "Embryonics + immunotronics: a bio-inspired approach to fault tolerance". In Proc. of NASA/DoD Workshop on Evolv. Hardw., pages 215–223, 2000.
10. C. S. Calude and G. Paun. "Computing with Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing". Taylor & Francis, 2001.
11. R. Cavill, S. Smith, and A. Tyrrell. "Multi-Chromosomal Genetic Programming". In Proc. Genetic and Evolutionary Computation Conference (GECCO 2005), pages 1753–1759, Washington DC, USA, June 2005.
12. A. Deckard and H. M. Sauro. "Preliminary Studies on the In Silico Evolution of Biochemical Networks". ChemBioChem, 5(10):1423–1431, 2004.
13. A. K. Dewdney. "Recreational Mathematics – Core Wars". Scientific American, May 1984. See also <http://www.koth.org/> and <http://www.corewars.org/>.
14. P. Dittrich. "Chemical Computing". In Unconventional Programming Paradigms (UPP 2004), Springer LNCS 3566, pages 19–32, 2005.
15. A. Eiben and J. Smith. "Introduction to Evolutionary Computing". Springer, 2003.
16. C. Ferreira. "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems". Complex Systems, 13(2):87–129, 2001.
17. J. A. Foster. "Evolutionary Computation". Nature Reviews Genetics, pages 428–436, June 2001.
18. R. A. Freitas Jr. and R. C. Merkle. "Kinematic Self-Replicating Machines". Landes Bioscience, Georgetown, TX, USA, 2004. available online <http://www.molecularassembler.com/KSRM.htm>.
19. J. Holland. "Adaptation in Natural and Artificial Systems". MIT Press, 1992. First Edition 1975.
20. Y. Jin and J. Branke. "Evolutionary Optimization in Uncertain Environments - A Survey". IEEE Transactions on Evolutionary Computation, 9(3):303–317, June 2005.
21. J. Koza. "Genetic Programming: On the Programming of Computers by Means of Natural Selection". MIT Press, 1992.
22. S. Kumar and P. J. Bentley. "Computational Embryology: Past, Present and Future". In Ghosh and Tsutsui, editors, Advances in Evolutionary Computing, Theory and Applications, pages 461–478. Springer, 2003.
23. P. Kuo, W. Banzhaf, and A. Leier. "Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence". Biosystems, 85:177–200, 2006.
24. W. B. Langdon and R. Poli. "Foundations of Genetic Programming". Springer, 2002.
25. A. Leier, P. D. Kuo, W. Banzhaf, and K. Burrage. "Evolving Noisy Oscillatory Dynamics in Genetic Regulatory Networks". In Proc. 9th European Conference on Genetic Programming, P. Collet, M. Tomassini, M. Ebner, S. Gustafson, A. Ekárt (Eds.) Springer LNCS 3905, pages 290–299, Budapest, Hungary, April 2006.
26. S. Luke and L. Panait. "Fighting Bloat With Nonparametric Parsimony Pressure". In Parallel Problem Solving from Nature (PPSN VII), LNCS 2439, pages 411–421, 2002.

27. Z. Manna and R. Waldinger. “*Fundamentals of Deductive Program Synthesis*”. IEEE Transactions on Software Engineering, 18(8):674 – 704, August 1992.
28. N. Matsumaru, P. S. di Fenizio, F. Centler, and P. Dittrich. “*On the Evolution of Chemical Organizations*”. In Proc. 7th German Workshop on Artificial Life, pages 135–146, 2006.
29. J. F. Miller and P. Thomson. “*Cartesian Genetic Programming*”. In R. P. et al., editor, Genetic Programming, Proceedings of EuroGP’2000, volume 1802 of LNCS, pages 121–132, Edinburgh, April 2000.
30. S. Nolfi and D. Floreano. “*Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*”. MIT Press, 2000.
31. M. O’Neill and C. Ryan. “*Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*”. Kluwer Academic Publishers, 2003.
32. C. Ortega-Sanchez, D. Mange, S. Smith, and A. Tyrrell. “*Embryonics: a bio-inspired cellular architecture with fault-tolerant properties*”. Genetic Programming and Evolvable Machines, 1:187–215, 2000.
33. G. Paun. “*Computing with Membranes*”. Journal of Computer and System Sciences, 61(1):108–143, 2000.
34. W. Piaseczny, H. Suzuki, and H. Sawai. “*Chemical Genetic Programming - The Effect of Evolving Amino Acids*”. In Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference (GECCO 2004), July 2004.
35. R. Poli. “*Parallel Distributed Genetic Programming*”. In New Ideas in Optimization, chapter 27, pages 403–431. McGraw-Hill, Maidenhead, Berkshire, England, 1999.
36. R. L. Probert and K. Saleh. “*Synthesis of Communication Protocols: Survey and Assessment*”. IEEE Transactions on Computers, 40(4):468 – 476, April 1991.
37. L. Prodan, G. Tempesti, D. Mange, and A. Stauffer. “*Embryonics: artificial stem cells*”. In Proc. of ALife VIII, pages 101–105, 2002.
38. O. G. Selfridge. “*Learning and Education: A Continuing Frontier for AI*”. IEEE Intelligent Systems, 21(3), May-June 2006.
39. N. Sharples. “*Evolutionary Approaches to Adaptive Protocol Design*”. PhD dissertation, University of Sussex, UK, August 2001.
40. N. Sharples and I. Wakeman. “*Protocol construction using genetic search techniques*”. In Real-World Applications of Evolutionary Computing – EvoWorkshops 2000, Springer LNCS 1803, Edinburgh, Scotland, April 2000.
41. M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Perez-Uribe, and A. Stauffer. “*A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-Inspired Hardware Systems*”. IEEE Transactions on Evolutionary Computation, 1(1), April 1997.
42. L. Spector, J. Klein, and M. Keijzer. “*The Push3 execution stack and the evolution of control*”. In Proc. Genetic and Evolutionary Computation Conference (GECCO 2005), pages 1689–1696, 2005.
43. L. Spector and A. Robinson. “*Genetic Programming and Autoconstructive Evolution with the Push Programming Language*”. Genetic Programming and Evolvable Machines, 3(1):7–40, 2002.
44. L. Spector and K. Stoffel. “*Automatic Generation of Adaptive Programs*”. In P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, editors, Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From animals to animats 4, pages 476–483, Cape Code, USA, 9-13 September 1996. MIT Press.
45. L. Spector and K. Stoffel. “*Ontogenetic Programming*”. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, Genetic Programming 1996: Proceedings of the First Annual Conference, pages 394–399, Stanford University, CA, USA, 28–31 July 1996. MIT Press.
46. K. O. Stanley and R. Miikkulainen. “*A taxonomy for artificial embryogeny*”. Artif. Life, 9:93–130, 2003.
47. R. Sterritt, M. G. Hinchey, J. L. Rash, W. Truskowski, C. Rouff, and D. Gracanic. “*Towards Formal Specification and Generation of Autonomic Policies*”. In 1st IFIP Workshop on Trusted and Autonomic Ubiquitous and Embedded Systems, Nagasaki, Japan, December 2005.
48. I. Tanev and K. Yuta. “*Epigenetic Programming: an Approach of Embedding Epigenetic Learning via Modification of Histones in Genetic Programming*”. In Proc. Congress on Evolutionary Computation (CEC), pages 2580–2587, 2003.
49. G. Tempesti, D. Mange, and A. Stauffer. “*Bio-inspired computing architectures: the Embryonics approach*”. In Proc. of IEEE CAMP, 2005.
50. J. Timmis, M. Amos, W. Banzhaf, and A. Tyrrell. “*Going back to our Roots: Second Generation Biocomputing*”. International Journal on Unconventional Computing, 2(4):349–382, 2006.
51. J. Weeds, B. Keller, D. Weir, I. Wakeman, J. Rimmer, and T. Owen. “*Natural Language Expression of User Policies in Pervasive Computing Environments*”. In Proc. LREC Workshop on Ontologies and Lexical Resources in Distributed Environments (OntoLex), 2004.
52. C. W.G.Lasarczyk and W. Banzhaf. “*An Algorithmic Chemistry for Genetic Programming*”. In Proc. 8th European Conference on Genetic Programming, M. Keijzer, A. Tettamanzi, P. Collet, M. Tomassini, J. van Hemert (Eds.) Springer LNCS 3447, pages 1–129, Lausanne, Switzerland, April 2005.
53. L. Yamamoto. “*Code Regulation in Open Ended Evolution*”. In Ebner et al., editor, Proceedings of the 10th European Conference on Genetic Programming (EuroGP 2007), volume 4445 of LNCS, pages 271–280, Valencia, Spain, April 2007. poster presentation.
54. L. Yamamoto and C. Tschudin. “*Experiments on the Automatic Evolution of Protocols using Genetic Programming*”. In Proc. 2nd Workshop on Autonomic Communication (WAC), pages 13–28, Athens, Greece, October 2005.

Evolutionary Games

Hamidou Tembine¹, Eitan Altman², Yezekael Hayel¹, Rachid El-Azouzi¹

¹ Université d'Avignon et des Pays de Vaucluse,
Laboratoire d'Informatique,
F-84911 Avignon, France

tembine@ieee.org, yezekael.hayel@univ-avignon.fr, Rachid.Elazouzi@univ-avignon.fr

² Institut National de Recherche en Informatique et Automatique,
INRIA Sophia Antipolis,
F-06902 Sophia Antipolis, France
eitan.altman@sophia.inria.fr

Abstract. Evolutionary games have developed in biological sciences to study equilibrium behaviour (called Evolutionary Stable Strategies – ESS) between large populations. ESS are more adapted than the standard Nash equilibrium to predict the evolution of large homogeneous or heterogeneous populations. While rich theoretical foundations of evolutionary games allow biologist to explain past and present evolution and predict future evolution, it can be further used in Engineering to architect evolution. In this paper, we introduce evolutionary games and present some useful related concepts. Our goal is to highlight the potential use of evolutionary games in networking. We present the challenge of architecting the evolution: we propose some guidelines for designing a framework that supports evolution of protocols and services.

1 Introduction

The evolutionary games formalism is a central mathematical tool developed by biologists for predicting population dynamics in the context of *interactions between populations*. This formalism identifies and studies two concepts: The ESS (for *Evolutionarily Stable Strategy*), and evolutionary game dynamics (such as the *Replicator Dynamics* [23]).

The ESS is characterized by a property of robustness against invaders (mutations). More specifically,

- (i) if an ESS is reached, then the proportions of each population do not change in time.
- (ii) at ESS, the populations are immuned from being invaded by other small populations. This notion is stronger than Nash equilibrium in which it is only requested that a single user would not benefit by a change (mutation) of its behaviour.

ESS has first been defined in 1972 by M. Smith [6], who further developed it in his seminal text *Evolution and the Theory of Games* [7], followed shortly by Axelrod's famous work [3].

Although ESS has been defined in the context of biological systems, it is highly relevant to engineering as well (see [8]).

Recently, evolutionary game theory has gained interest among social scientists [16]. In the biological context, the replicator dynamics is a model for the change of the size of the population(s) as biologist observe, whereas in engineering, we can go beyond characterizing and modelling existing evolution. The evolution of protocols can be engineered by providing guidelines or regulations for the way to upgrade existing ones and in determining parameters related to deployment of new protocols and services. In doing so we may wish to achieve adaptability to changing environments (growth of traffic in networks, increase of speeds or of congestion) and yet to avoid instabilities that could otherwise prevent the system to reach an ESS.

The first objective in introducing evolutionary games is to provide a framework to describe and predict evolution of protocols and of architecture of networks in a context of competition between two types of behaviours: aggressive and peaceful. To study this, and to illustrate the properties of evolutionary games, we present the well known Hawk and Dove Game. We identify cases in which at ESS only one population prevails (ESS in pure strategies) and others in which an equilibrium between several

population types is obtained. In the latter case, the framework of evolutionary games allows us to compute the proportion of each population at equilibrium.

The second objective of the paper is to present the notion of replicator dynamics. While this notion is at the heart of modelling and predicting evolution in biology, it can also provide a framework in networking for controlling evolutionary dynamics such as changing or upgrading network architecture and protocols, and evolution of services. We illustrate this point by showing the impact of the choice of some parameters defining the replicator dynamics on its convergence and stability.

Two applications to network protocols are provided to illustrate the usefulness of evolutionary games. The first application focuses on a Multiple-Access protocol and the second on the competition between various variants of transport protocols over wireless internet, and their evolution.

2 The Framework

Consider a large population of players. Each individual needs occasionally to take some action (such as power control decisions, or forwarding decision). We focus on some (arbitrary) tagged individual. Occasionally, the action of some N (possibly random number of) other individuals interact with the action of that individual (e.g. other neighbouring nodes transmit at the same time). In order to make use of the wealth of tools and theory developed in the biology literature, we shall often restrict, as they do, to interactions that are limited to pairwise, i.e. to $N = 1$. This will correspond to networks operating at light loads, such as sensor networks that need to track some rare events such as the arrival at the vicinity of a sensor of some tagged animal.

We define by $J(p, q)$ the expected payoff for our tagged individual if it uses a strategy p when meeting another individual and when the other customer adopts independently the strategy q . This payoff is called "fitness" and strategies with larger fitness are expected to propagate faster in a population.

We assume that there are K pure strategies: (s_1, \dots, s_k) . A strategy of an individual is a probability distribution over the pure strategies. An equivalent interpretation of strategies is obtained by assuming that individuals choose pure strategies and then the probability distribution represents the fraction of individuals in the population that choose each strategy. Note that J is linear in p and in q . Indeed, it is given by

$$J(p, q) = \sum_{i=1}^K \sum_{j=1}^K p(s_i) q(s_j) J(s_i, s_j).$$

3 Evolutionary Stable Strategies

Suppose that the whole population uses a strategy q and that a small fraction ε (called "mutations") adopts another strategy p . Evolutionary forces are expected to select against p if

$$J(q, \varepsilon p + (1 - \varepsilon)q) > J(p, \varepsilon p + (1 - \varepsilon)q) \quad (1)$$

A strategy q is said to be ESS if for every $p \neq q$ there exists some $\bar{\varepsilon}_y > 0$ such that (1) holds for all $\varepsilon \in (0, \bar{\varepsilon}_y)$.

In fact, we expect that if for all $p \neq q$,

$$J(q, q) > J(p, q) \quad (2)$$

then the mutations fraction in the population will tend to decrease (as it has a lower reward, meaning a lower growth rate). q is then immune to mutations. If it does not but if still the following holds,

$$J(q, q) = J(p, q) \text{ and } J(q, p) > J(p, p) \quad \forall p \neq q \quad (3)$$

then a population using q are "weakly" immune against a mutation using p since if the mutant's population grows, then we shall frequently have individuals with strategy q competing with mutants; in such cases, the condition $J(q, p) > J(p, p)$ ensures that the growth rate of the original population exceeds that

of the mutants. A strategy is ESS if and only if it satisfies (2) or (3), see [29, Proposition 2.1] or [5, theorem 6.4.1, page 63].

The conditions on ESS can be related to and interpreted in terms of Nash equilibrium in a matrix game. The situation in which an individual, say player 1, is faced with a member of a population in which a fraction p chooses strategy A is then translated to playing the matrix game against a second player who uses mixed strategies (randomizes) with probabilities p and $1 - p$, resp. The central model that we shall use to investigate protocol and service evolution is introduced in the next Subsection along with its matrix game representation.

4 The Hawk and Dove (HD) Game



Consider a large population of animals. Occasionally two animals find themselves in competition on the same piece of food. An animal can adopt an aggressive behavior (Hawk) or a peaceful one (Dove). The matrix in Fig. 1 presents the fitness of player I (some arbitrary player) associated with the possible outcomes of the game as a function of the actions taken by each one of the two players. We assume a symmetric game so the utilities of any animal (in particular of player 2) as function of its actions and those of a potential adversary (in particular of player 1), are the same as those player 1 has in Figure 1. The utilities represent the following:

- An encounter D–D results in a peaceful, equal-sharing of the food which translates to a fitness of 0.5 to each player.
- An encounter H–H results in a fight in which with equal chances one or the other player obtains the food but also, in which there is a positive probability for each one of the animals to be wounded. The fitness of each player is $0.5-d$, where the 0.5 term is as in the D–D encounter and the $-d$ term represents the expected loss of fitness due to being injured.
- An encounter H–D or D–H results in zero fitness to the D and in one unit of utility for the H that gets all the food without fight.

Classification of equilibria in the HD game

A more general description of H–D games is available in [26,24]. One can indeed think of other scenarios that are not covered in the original H–D game, such as the possibility of a Hawk to find the Dove, in a H–D encounter, more delicious than the food they compete over.

In the generalized version [14] of the HD game given in Figure 2, if $A_{11} > A_{21}$ and then (H, H) is the unique Nash equilibrium. If $A_{11} < A_{21}$ and then the strategies (H, D) and (D, H) are pure Nash equilibria. $a = \frac{u}{u+v}$ is the unique interior Nash equilibrium where $A_{ij} = J(i, j)$, $i, j \in \{H, D\}$, $u = A_{12} - A_{22}$, $v = A_{21} - A_{11}$

		Player II	
		H	D
Pl. I	H	$0.5 - d$	1
	D	0	0.5

Fig. 1. A H–D game in matrix form

		Player II	
		H	D
Pl. I	H	A11	A12
	D	A21	A22

Fig. 2. Generalized H–D game

Remark 1. (i) Note that there are no settings of parameters for which the pure strategy D is an ESS in the H–D game (or in its generalized version).

(ii) In case 2 above, the strategies (H, D) and (D, H) are pure Nash equilibria in the matrix game. Being asymmetric, they are not candidates for being an ESS according to our definition. There are however contexts in which one obtains non-symmetric ESS, in which case they turn out to be ESS.

5 Evolution: Replicator Dynamics

We introduce here the replicator dynamics which describes the evolution in the population of the various strategies. In the replicator dynamics, the share of a strategy in the population grows at a rate equal to the difference between the average payoff of that strategy and the average payoff of the population as a whole (see [10]).

To be more precise, consider the case of a finite number N of strategies. Let \mathbf{x} be the N dimensional vector whose i th element x_i is the population share of strategy i . Thus $\sum_i x_i = 1$ and $x_i \geq 0$. Below we denote by $J(i, k)$ the expected payoff (or the fitness) for a player using a strategy i when it encounters a player with strategy k . With some abuse of notation we define $J(i, \mathbf{x}) = \sum_j J(i, j)x_j$. Then the replicator dynamics is defined as

$$\begin{aligned} \dot{x}_i(t) &:= \frac{dx_i(t)}{dt} = x_i K \left(J(i, \mathbf{x}) - \sum_j x_j J(j, \mathbf{x}) \right) \\ &= x_i K \left(\sum_j x_j J(i, j) - \sum_j \sum_k x_j J(j, k)x_k \right) \end{aligned} \quad (4)$$

where K is some positive constant.

Note: summing the right hand side over i , it is seen that the right hand side is zero. This is compatible with the fact that we study here the share of each strategy rather than the size of the population that uses each one of the strategies.

The replicator dynamics has been used for describing the evolution of road traffic congestion in which the fitness is determined by the strategies chosen by all drivers [11]. It has also been studied in the context of the association problem in wireless networks in [12].

6 Replicator Dynamics with Delay

In (5), the fitness of strategy i at time t has an instantaneous impact on the rate of growth of the population size that uses it. An alternative more realistic model for replicator dynamic would have some delay: the fitness acquired at time t will impact the rate of growth τ time later. This gives the following dynamics:

$$\dot{x}_i(t) = x_i(t) K \left(J(i, \mathbf{x}(t - \tau)) - \sum_j x_j(t) J(j, \mathbf{x}(t - \tau)) \right) \quad (5)$$

$$= x_i(t)K \left(\sum_j x_j(t-\tau)J(i,j) - \sum_{j,k} x_j(t)J(j,k)x_k(t-\tau) \right)$$

where K is some positive constant. We should mention that the delay τ represents a time scale much slower than the physical (propagation and queueing) delays, and in the context of evolution of protocols, it is related to the time scale of (i) switching from the use of one protocol to another (ii) upgrading protocols.

7 Other Evolutionary Models

There is a large number of population dynamics other than the replicator dynamics which have been used in the context of non-cooperative games. Examples are the Brown – von Neumann – Nash [4] dynamics, the fictitious play dynamics and gradient methods [9]. See also [10,28].

We finally mention the logistic differential equation:

$$\frac{dx(t)}{dt} = Kx(t) \left(1 - \frac{x(t)}{N} \right).$$

It is frequently used in population dynamics. As an example, it is used in epidemiology to describe the fraction x of the population that has been infected. It does not involve decision or game.

8 Application to Multiple Access Protocols

To illustrate the relevance to networking, we focus on a simple model with one population of users where each individual has a choice between two strategies. We assume as usual that the interactions between the strategies are manifested through many local interactions between pairs of users. We shall use the standard representations of the evolutionary game as a two players matrix game representing the expected fitness obtained in an interaction between two individuals; the expectation is with respect to the fraction of the population that uses each strategy. We assume that time delays are not necessary symmetric.

The Model

Consider a large population of mobile terminals in ad hoc network and assume that the density of the network is low, so that if a terminal attempts transmission one can neglect the probability of interference from more than one other mobile (called "neighbour").

We assume the mobiles move frequently and they have a packet to send in each time slot. A mobile decide to transmit or not a packet to a receiver when they are within transmission range of each other. Interference occurs as in the ALOHA protocol: if more than one two neighbours transmit a packet at the same time then there is a collision. The Multiple Access Game is a symmetric nonzero-sum game, but the users have to share a common resource, the wireless medium, instead of providing it. Assume that the users use pure strategy. When there are two users Player I and Player II who want to send some packets to their receivers R1 and R2 using a shared medium. We assume that the users have a packet to send in each time slot and they can decide to transmit it or not. Suppose furthermore that Player I, Player II, R1 and R2 are in the power range of each other, hence their transmissions mutually interfere. Each of the users has two possible strategies: either transmit (T) or to stay quiet (S). If Player I transmits his packet, it incurs a transmission cost of $\Delta \in (0, 1)$ after a delay τ_T . The packet transmission is successful if Player II does not transmit (stays quiet) in that given time slot and its delay is τ_S , otherwise there is a collision. If there is no collision, Player I gets a reward of V (normalized to unit) from the successful packet transmission after the delay τ_T . The interaction is represented in figure 3.

Player I \ Player II	T	S
T	$-\Delta$	$V - \Delta$
S	0	0

Fig. 3. Representation of the symmetric Multiple Access Game.

Delay Impact on the Stability

The one-shot ALOHA game between two users has three Nash equilibria: (T, S) , (S, T) and $(1 - \Delta)T + \Delta S$. It is easy to show that the unique mixed equilibrium is an ESS, the pure equilibria are not symmetric. The replicator dynamic equation with asymmetric delays in the Multiple Access Game becomes

$$\dot{\xi}(t) = -K\xi(t)(1 - \xi(t)) [\xi(t - \tau_T) - 1 + \Delta] \quad (6)$$

where $\xi(t)$ proportion of individuals using the the strategy T at time t . The ESS $(1 - \Delta, \Delta)$ is asymptotically stable for the replicator dynamics given in 6 if $2K\Delta(1 - \Delta)\tau_T < \pi$ and not stable if $2K\Delta(1 - \Delta)\tau_T > \pi$. (a proof can be found in [28]).

The trajectories of the population using the strategy T , as a function of time is represented in Fig.4. We evaluate the stability varying the delay τ_T in the replicator dynamic. When the delay τ_T is large, the trajectory oscillates rapidly and the amplitude is seen to be greater than the equilibrium point and the system becomes is unstable. Note that stability condition is independent of τ_S .

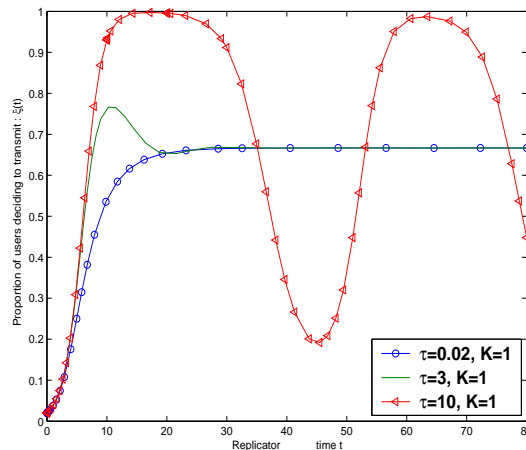


Fig. 4. Effect of τ_T on velocity and stability of replicator dynamic. $\Delta = 1/3$

9 Application to Transport Protocols

We summarize below a study in [31] based on evolutionary games of competition between various variants of the TCP transport protocols in a wireless context, and their predicted evolution.

Background When transferring data between nodes, flow control protocols are needed to regulate the transmission rates so as to adapt to the available resources. A connection that loses data units has to retransmit them later. In the absence of adaptation to the congestion, the ongoing transmissions along with the retransmissions can cause increased congestion in the network resulting in losses and further retransmissions by this and/or by other connections. This type of phenomenon, that leads to several 'congestion collapses' [32], motivated the evolution of the Internet transport protocol, TCP, to a protocol that reduces dramatically its throughput upon congestion detection.

There are various versions of the TCP protocol among which the mostly used one is New-Reno. The degree of 'aggressiveness' varies from version to version. The behavior of New-Reno is approximately

AIMD (Additive Increase Multiplicative Decrease): it adapts to the available capacity by increasing the window size in a linear way by α packets every round trip time and when it detects congestion it decreases the window size to β times its value. The constants α and β are 1 and 1/2, respectively, in New Reno.

In last years, more aggressive TCP versions have appeared, such as HSTCP (High Speed TCP) [33] and Scalable TCP [34]. HSTCP can be modelled by an AIMD behavior where α and β are not constant any more : α and β have minimum values of 1 and of 1/2, resp. and both increase in the window size. Scalable TCP is an MIMD (Multiplicative Increase Multiplicative Decrease) protocol, where the window size increases exponentially instead of linearly and is thus more aggressive. Versions of TCP which are less aggressive than the New-Reno also exist, such as Vegas [35].

Several researchers have analysed the performance of networks in which various transport protocols coexist, see [36,37,38,39,40,41]. In all these papers, the population size using each type of protocol is fixed.

Some papers have already considered competition between aggressive and well behaved congestion control mechanisms within a game theoretic approach. Their conclusions in a wireline context was that if connections can choose selfishly between a well behaved cooperative behavior and an aggressive one then the Nash equilibrium is obtained by all users being aggressive and thus in a congestion collapse [42,43].

Our approach yields qualitative results, stronger than those obtained through the traditional Nash equilibrium concept adopted in these references. It allows in particular to study the evolution to the equilibrium, and to obtain a sharper characterization of the equilibrium as being robust not only against a single user deviation but also against deviations of a whole (small) fraction of the population.

By casting the problem in our framework of the Hawk and Dove evolutionary game, we shall be able to predict whether a given version of TCP is expected to dominate others (ESS in pure strategies, which means that some versions of TCP would disappear) or whether several versions would co-exist. This would depends also on the network context: an aggressive version of TCP that may dominate in a wireline context may loose its dominance in a wireless network. Indeed, an aggressive TCP may generate higher packet loss rate than other less aggressive versions. These are evaluated more severely in a wireless environment since they represent energy inefficiency which is costly in that environment.

During the last few years, many researchers have been studying TCP performance in terms of energy consumption and average goodput within wireless networks [44,45]. Via simulation, the authors show that the TCP New-Reno can be considered as well performing within wireless environment among all other TCP variants and allows for greater energy savings. Indeed, a less aggressive TCP, as TCP New-Reno, may generate lower packet loss than other aggressive TCP. By using the HD game, we show the same behavior of TCP variants.

The Model

We consider two populations of connections, all of which use AIMD TCP. A connection of population i is characterized with a linear increase rate α_i and a multiplicative decrease factor β_i . Let $x_i(t)$ be the transmission rate of connection i at time t . We consider the following simple model for competition.

- The RTT (round trip times) are the same for all connections.
- There is light traffic in the system in the sense that a connection either has all the resources its needs or it shares the resources with one other connection. (If files are large then this is a light regime in terms of number of connections but not in terms of workload).
- Losses occur whenever the sum of rates reaches the capacity C : $x_1(t) + x_2(t) = C$.
- Losses are synchronized: when the combined rates attain C , both connections suffer from a loss. This synchronization has been observed in simulations for connections with RTTs close to each other [1]. The rate of connection i is reduced by the factor $\beta_i < 1$.
- As long as there are no losses, the rate of connection i increases linearly by a factor α_i .

We say that a TCP connection i is more aggressive than a connection j if $\alpha_i \geq \alpha_j$ and $\beta_i \geq \beta_j$. Let $\bar{\beta}_i := 1 - \beta_i$.

HD game: Throughput-loss Tradeoff between two AIMD TCP

In wireline, the utility related to file transfers is usually taken to be the throughput, or a function of the throughput (e.g. the delay). It does not explicitly depend on the loss rate. This is not the case in wireless context. Indeed, since TCP retransmits lost packets, losses present energy inefficiency. Since energy is a costly resource in wireless, the loss rate is included explicitly in the utility of a user through the term representing energy cost. We thus consider fitness of the form $J_i = Thp_i - \lambda R$ for connection i ; it is the difference between the throughput Thp_i and the loss rate R weighted by the so called tradeoff parameter, λ , that allows us to model the tradeoff between the valuation of losses and throughput in the fitness. We now proceed to show that our competition model between aggressive and non-aggressive TCP connections can be formulated as a HD game. We study how the fraction of aggressive TCP in the population at (the mixed) ESS depends on the tradeoff parameter λ .

It is easily seen that the share of the bandwidth (just before losses) of a user is increasing in its aggressiveness. The average throughput of connection 1 is given

$$Thp_1 = \frac{1 + \beta_1}{2} \times \frac{\alpha_1 \bar{\beta}_2}{\alpha_1 \bar{\beta}_2 + \alpha_2 \bar{\beta}_1} \times C.$$

The average loss rate of connection 1 is the same as that of connection 2 and is given by

$$R = \left(\frac{\alpha_1}{\beta_1} + \frac{\alpha_2}{\beta_2} \right) \frac{1}{C}$$

Let H corresponds to (α_H, β_H) and D to (α_D, β_D) such that $\alpha_H \geq \alpha_D$ and $\beta_H \geq \beta_D$. Since the loss rate for any user is increasing in $\alpha_1, \alpha_2, \beta_1, \beta_2$ it then follows under choosing the parameters $\lambda, \alpha_i, \beta_i, i = 1, 2$, that the utility that describes a tradeoff between average throughput and the loss rate leads to the HD structure. Hence, the game has only one ESS which depends on the tradeoff parameter λ and the ESS is polymorphic i.e the two versions of TCP would coexist.

A stability analysis is carried out in [31] and an oscillatory behavior is identified in the non-stable case.

10 Conclusions

We presented in this paper the foundations of evolutionary games. We highlighted its advantages with respect to traditional games: the more robust equilibrium notion called ESS, and the evolutionary (replicator) dynamics that allows to predict the propagation of new strategies in a population. We illustrated these points through various applications.

References

1. O. Ait-Hellal, E. Altman, D. Elouadghiri, M. Erramdani, N. Mikou, "Performance of TCP/IP: the case of two Controlled Sources", ICC'97, Cannes, France, November 19-21, 1997.
2. Jan Alboszta and Jacek Mięksiz. *Stability and evolutionary stable strategies in discrete replicator dynamics with delay*.
3. Axelrod, R. (1984) *The Evolution of Cooperation*. New York: Basic Books.
4. G. Brown and J. Von Neumann. Solutions of games by differential equations. In H. Kuhn and A. Tucker, editors, *Contributions to the Theory of Games I*, *Annals of Mathematics Studies* 24, pages 73-79. Princeton University Press, 1950.
5. Josef Hofbauer and Karl Sigmund, " *Evolutionary Games and Population Dynamics*", Cambridge University Press 1998.
6. M. Smith, 1972. "Game Theory and the Evolution of Fighting." In John Maynard Smith, *On Evolution* (Edinburgh: Edinburgh University Press), pp.8-28.
7. M. Smith, *Evolution and the Theory of Games*, Cambridge University Press, Cambridge, UK, 1982.
8. T. L. Vincent and T. L. S. Vincent, "Evolution and control system design", *IEEE Control Systems Magazine*, Vol 20 No. 5, pp 20-35, Oct. 2000.
9. J.B. Rosen. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica*, 33:153-163, 1965.
10. L. Samuelson, "Evolution and Game Theory", *The journal of Economics Perspectives*, Vol 16 No 2, pp 47-66, 2002.
11. W.H. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97:81=108, 2001.

12. S. Shakkottai, E. Altman and A. Kumar, "The Case for Non-cooperative Multihoming of Users to Access Points in IEEE 802.11 WLANs", *IEEE Infocom*, 2006.
13. Tao, Y. and Wang, Z., 1997. Effect of time delay and evolutionarily stable strategy. *J. Theor. Biol.* 187, 111-116.
14. Bruce S. Cushing. "When a Hawk can damage a Dove : An extension of Game Theory", *J. Theor. Biol.*, 173-176, 1995.
15. Charles A. Desoer and Yung-Terng Wang. "On the generalized nyquist stability criterion", *IEEE Transactions on Automatic Control*, vol. AC-25, NO. 2, april 1980.
16. Daniel Friedman. On economic applications of evolutionary game theory. *Journal of Evolutionary Economics*, 8(1):15–43, 1998.
17. S.J. Mason. A comment on dr vazsonyi's paper: "a generalized nyquist's stability criteria", may,20. 1949.
18. P. K. Stevens. "A generalization of the nyquist stability criterion", *IEEE Transactions on Automatic Control*, vol AC-26, no. 3, june, 1981.
19. K. Gopalsamy. "Stability and Oscillation in Delay Differential Equations of Population Dynamics", Kluwer Academic Publishers, London, 1992.
20. R. D. Driver, D. W. Sasser, and M. L. Slater. "The equation $x'(t) = ax(t) + bx(t - \tau)$ with " Small" Delay", *The American Mathematical Monthly*, Vol.80, No9, pp.990-995, 1973.
21. Leonid Berezansky and Elena Braverman. On stability of some linear and nonlinear delay differential equations. *J. Math. Anal. Appl.*, Vol. 314, No. 2 (2006), pages 391-411, 2006.
22. Khalil, H.K.: Nonlinear Systems. Prentice-Hall, Upper Saddle River, NJ (2002)
23. J. Hofbauer and K. Sigmund, "Evolutionary game dynamics", *American Mathematical Society*, Vol 40 No. 4, pp. 479–519, 2003.
24. A. I. Houston and J. M. McNamara, "Evolutionarily stable strategies in the repeated hawk-dove game", *Behavioral Ecology*, pp. 219–227, 1991.
25. J. M. McNamara, "The policy which maximizes long-term survival of an animal faced with the risks of starvation and predation", *Advances of Applied Probability*, 22, 295–308,1990.
26. J. M. McNamara, S. Merad and E. J. Collins, "The hawk-dove game as an average cost problem", *Advances of Applied Probability*, volume 23, pp 667–682, 1991.
27. V. Mhatre and C. Rosenberg, "Energy and cost optimizations in wireless sensor networks: A survey," in the 25th Anniversary of GERAD, Kluwer Academic Publishers, Jan 2004.
28. H. Tembine, E. Altman and R. El-Azouzi. Delayed Evolutionary Game Dynamics applied to the Medium Access Control, in Proc. IEEE MASS 2007.
29. J.W. Weibull. *Evolutionary Game Theory*. Cambridge, MA: MIT Press, 1995.
30. Tao Yi and Wang Zuwang. *Effect of Time Delay and Evolutionary Stable Strategy*. *J. theor. Biol*, 187, 111-116, 1997.
31. H. Tembine, E. Altman, R. ElAzouzi and Y. Hayel, "Evolutionary games for predicting the evolution and adaptation of wireless protocols", technical report, University of Avignon, 2008.
32. Van Jacobson "Congestion Avoidance and Control", *SIGCOMM '88*, Stanford, CA, Aug. 1988.
33. E. Souza and D.A. Agarwal. "A HighSpeed TCP study: Characteristics and deployment issues", Technical Report LBNL-53215, Lawrence Berkeley National Laboratory, 2002.
34. T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks". *Computer Communication Review* 32(2), April 2003.
35. Brakmo L.S., O'Malley S., and L.L. Peterson. "Tcp vegas: New techniques for congestion detection and avoidance", *Computer Communication Review*, Vol. 24, No. 4, pp. 24-35, Oct., 1994.
36. A. Tang, J. Wang, S. Low, and M. Chiang. "Equilibrium of heterogeneous congestion control: Existence and uniqueness", *IEEE/ACM Transactions on Networking*, October 2007.
37. T. Bonald, "Comparison of TCP Reno and TCP Vegas: Efficiency and Fairness", *Perform. Eval.* 36-37(1-4): 307-332 (1999).
38. O Ait-Hellal and E. Altman. "Analysis of TCP Vegas and TCP Reno", *Telecommunication Systems*, 15:381–404, 2000.
39. E. Altman, K. Avrachenkov, and B. Prabhu. "Fairness in MIMD congestion control algorithms", *Telecommunication Systems*, 30(4):387–415.
40. L. Lopez, A. Fernandez and V. Cholvi. "A Game Theoretic Analysis of Protocols Based on Fountain Codes". *IEEE ISCC'2005*, June 2005
41. Konstantin Avrachenkov, Moller Niels, Chadi Barakat and Eitan Altman, "Inter-protocol fairness between TCP New Reno and TCP Westwood+", *NGI*, 2007.
42. R. Garg, A. Kamra, V. Khurana, A game-theoretic approach towards congestion control in communication networks *SIGCOMM Computer Communication Review* archive 32(3) (July 2002) 47 - 61
43. L. Lopez, G. Rey, A. Fernandez and S. Paquelet, "A mathematical model for the TCP tragedy of the commons", *Theoretical Computer Science*, 343, pp. 4–26, 2005.
44. H. Singh, S. Singh, "Energy consumption of TCP Reno, New Reno and SACK in multi-hop wireless networks, *ACM SIGMETRICS*, Jun. 2002.
45. M. Zorzi and R. Rao, "Energy efficiency of TCP in a local wireless environment, *Mobile Networks and Applications*, Vol. 6, No. 3, Jul. 2001

Activation-Inhibition Mechanisms for Distributed Coordination

Daniele Miorandi¹, Karina M. Gomez¹, Lidia Yamamoto²

¹ CREATE-NET

via alla Cascata 56/D

Povo, Trento – 38123, Italy

daniele.miorandi, karina.gomez@create-net.org

² Computer Science Department, University of Basel

Bernoullistrasse 16, CH–4056 Basel, Switzerland

lidia.yamamoto@unibas.ch

Abstract. In this chapter we review activator–inhibitor mechanisms and their application to distributed coordination problems in networking and computing. We first introduce the underlying mechanisms, based on chemical reaction and diffusion of molecules. We then present mathematical models used for describing activator–inhibitor mechanisms and the patterns that they can give rise to. We finally discuss applications to networking and computing problems, whereby activator–inhibitor mechanisms are engineered to design autonomous distributed coordination schemes.

1 Introduction

One of the most intriguing processes in biology is *morphogenesis*, the formation of a complex multicellular organism from a single and simple egg. Morphogenesis relies on the diffusion of chemicals — *morphogens* — that are used as signals to trigger the differentiation of cells into various cell types (blood, tissues, organs), and the formation of shapes for the various body structures (e.g. fingers, blood vessels, eyes). This process involves symmetry–breaking steps —during the developmental phase— which have no external triggers but are internally generated. Since chemicals tend to diffuse in a homogeneous way in all directions, the ability to create asymmetric structures has attracted much interest from both biologists and mathematicians. Further, the structures formed through morphogenesis often present a high level of robustness with respect to perturbations in both the system and the environmental conditions.

In his famous 1952 paper [24], Alan Turing provided a pioneering mathematical model for morphogenesis. His model offered the first potential explanation for pattern formation phenomena observed in the biological world, such as spots and stripes on animal coating, such as zebras, leopards, etc. Today, these patterns are referred to as *Turing patterns*. Although morphogenesis is actually much more complicated than predicted by Turing, the basic idea remains still valid.

Turing’s model of morphogenesis is based on *reaction-diffusion*, a system of partial differential equations combining the diffusion of chemicals in space with their chemical reactions. Reaction-diffusion equations are widely used today to model various pattern formation mechanisms in biology, chemistry, physics, and in complex systems in general. Several types of reaction-diffusion systems are known to spontaneously form different kinds of patterns: the Belousov-Zhabotinsky reaction, the Gray-Scott model [20], activator-inhibitor models, the hypercycle in space [4], among others. Besides the spots and stripes already mentioned above, other typical patterns include waves, spirals, and self-replicating spots. Moreover, *reaction-diffusion computers* [2] are being proposed as unconventional models of computation that make use of patterns in chemical media to compute distributed algorithms.

In this chapter, we review one of the key morphogenetic mechanisms: *activator–inhibitor* dynamics. We then survey its usage as a design tool for achieving coordination in a distributed computing system.

Activation–inhibition models describe situations in which two competing processes take place over space and time. The first one (*activation*) tends to self–enhance the process within the local neighbourhood. A competing force (*inhibition*), weaker but with a longer spatial range, tends to decrease the activation effect in the surrounding space. Under certain conditions, the joint effect of such two forces

may give rise to asymmetric spatial distributions, resulting in patterns such as spots and stripes, resembling those found on the skin of animals.

In nature, activators and inhibitors are molecules that may diffuse over space. Activators trigger autocatalytic reactions that increase their own concentration (self-enhancement), but such effect has limited spatial range. On the other hand, activators also trigger chemical reactions that tend to (slightly) decrease the concentration of activators (inhibition). Such a “negative impact” process has a reduced intensity when compared to self-enhancement, but has much larger spatial range.

To illustrate the concept, we can consider the formation of dunes in desertic areas. If the area is totally flat, no matter in which direction the wind is blowing, no dunes can form. However, if there is even a very small wind shelter, sand will accumulate behind it. This deposit will lead to more sand accumulating (self-enhancement). At the same time the fact that sand gets stopped by the deposit reduces the amount of sand that can deposit in nearby areas (inhibition). The net result is the formation of a dune, i.e., an asymmetry in the spatial distribution of sand.

We can then consider a more abstract example, in particular a process whereby, at the equilibrium, a substance S is uniformly distributed over space. Now, let us perturb the system at a given point x by increasing the value of S . The inhibition effect will tend to lower the resulting system value over a wide surface. The activation effect will tend to raise the value nearby point x . If the activation value at x is larger than the inhibition one, the value at x will tend to increase. After a few iterations we will notice the emergence of a peak at x . This can be regarded as a positive feedback loop; its net effect will be the break of symmetry in the resulting distribution.

The fact that reaction-diffusion models are at the basis of pattern formation suggests that they could potentially be applied to the construction of self-organized coordination mechanisms, whereby a given pattern has to be achieved in order to let a system perform a given function. This is one of the basic ideas behind reaction-diffusion computers [2], but it can also be applied to algorithms running on conventional silicon hardware. Indeed, reaction-diffusion has already found applications in distributed systems and networking, and has been proposed as a general paradigm for the realization of a new class of highly distributed computing systems, called Amorphous Computing [1]. In this chapter we review these applications with focus on activator-inhibitor mechanisms.

The remainder of this chapter is organized as follows. In Sec. 2 we describe the role played by activator-inhibitor mechanisms in a variety of biological processes. In Sec. 3 we present a number of mathematical models developed, mostly by mathematical biologists, to describe the resulting effect of activator-inhibitor mechanisms. In Sec. 4 we survey related work on the application of activator-inhibitor mechanisms to the construction of coordination mechanisms in distributed computing systems. Sec. 5 concludes the chapter pointing out some promising applications and research directions.

2 Activation–Inhibition Mechanisms in Biology

Much progress has been achieved since Alan Turing’s pioneering model of morphogenesis in 1952 [24], which could explain animal coating patterns such as zebra stripes and leopard spots. Morphogenesis is an active area of research, with numerous potential applications in the medicine, biology and other domains [3,6,17,18].

In his book [17], Hans Meinhardt provides detailed descriptions of many pattern formation phenomena occurring in biology, together with their possible mathematical explanations. For instance, gradients of chemicals can provide positional information in an embryo, guiding cell differentiation and the formation of organism structures. Such gradients can also control the replication and regeneration of missing elements in a structure. Oscillating patterns can be used as clock information for various biological functions. Many explanations for such natural phenomena still remain hypothetical, and have not yet been demonstrated in practice. This is mainly due to the complexity of most biological processes such as the impact of gene expression on cell signalling phenomena leading to chemical diffusion and the reading of chemical signals to feedback the underlying gene regulatory machinery. Much remains to be done in order to fully explain and control the developmental process.

Besides the formation of skin patterns, activator-inhibitor models can explain many different morphogenetic phenomena, including the regular spacing of thorns on a cactus and feathers on a bird, the

regeneration of a shape after some damage, the production of sequences of repeated elements such as insect body segments, the assembly of photoreceptor cells in insect eyes, and the positioning of leaves in growing plants. Reticulated patterns such as a giraffe coat can be explained by a modified version of the activator-substrate model (see section 3.1) combined with a switching system.

In [17], the importance of autocatalysis and long-range inhibition is highlighted as a general mechanism of pattern formation based on activator-inhibitor, as will be discussed in more detail in the next section. Typically however, activator-inhibitor models lead to symmetric, spot-like patterns. The phenomenon of *lateral inhibition* offers an explanation for stripe patterns such as zebra coatings. Lateral inhibition is based on the saturation of autocatalysis in the neighbourhood of an area. Lateral inhibition can also contribute to the formation of tree or network-like structures such as dendritic connections in nerve cells, blood vessels, lungs, etc.

3 Mathematical Models

In this section we summarize a set of mathematical models developed to analyse activator–inhibitor mechanisms. Activator–inhibitor mechanisms are usually modelled by considering a number (in most cases two) of non-linear reaction–diffusion equations respecting certain monotonicity properties. One of the two functions (the activator a) increases the rate of both reactions. The other one (the inhibitor h) inhibits them. Under certain conditions on the relative magnitude of activation and inhibition effects, and on their ability to diffuse over space, this model can give rise to asymmetries in the resulting distribution [10].

A model of activator–inhibitor dynamics is presented in Turing’s seminal paper [24]. The main ideas at the basis of Turing’s model were the following ones:

- A symmetric steady state is always present. It is robust to symmetric perturbations, but not to asymmetric ones.
- As parameters in the model change, a qualitative change in the resulting solution happens.
- Interaction between a short–range activation and a long–range inhibition can lead to the formation of asymmetric spatial patterns.

Models for activation–inhibition mechanisms build on the reaction–diffusion basic equation [18]:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{f}(\mathbf{c}) + D\nabla^2 \mathbf{c}, \quad (1)$$

where \mathbf{c} is a vector–valued quantity (referring, e.g., to the concentration level of a given chemical), \mathbf{f} describes the reaction kinetics, D is a diagonal matrix representing the diffusion coefficients and ∇^2 is the Laplacian operator.

By specializing the model to two chemicals (other models are also possible, but this is the most widely used one), the following differential equations are obtained:

$$\frac{\partial a}{\partial t} = F(a, h) + D_a \nabla^2 a; \quad (2)$$

$$\frac{\partial h}{\partial t} = G(a, h) + D_h \nabla^2 h. \quad (3)$$

In order to give rise to non–trivial (non–homogeneous) solutions, both $F(\cdot, \cdot)$ and $G(\cdot, \cdot)$ have to be non–linear functions.

This was the basic model studied by Turing [24]. He noticed that if $D_a = D_h = 0$, i.e., no diffusion is present, then the process (a, h) converges (irrespective of the initial conditions) to a spatially uniform steady–state. The values assumed by a and h in steady–state would depend on both reaction kinetics $F(\cdot, \cdot)$ and $G(\cdot, \cdot)$ as well as on the initial values of the field. However, if $D_a \neq D_h$, a phenomenon called “diffusion–driven instability” could take place, leading to the emergence of non–uniform patterns.

Necessary and sufficient conditions for a system like that in Equations (2-3) to build non-uniform patterns can be given as follows [18]:

$$\frac{\partial F}{\partial a}(a_0, h_0) + \frac{\partial G}{\partial h}(a_0, h_0) < 0; \quad (4)$$

$$\frac{D_h}{D_a} \frac{\partial F}{\partial a}(a_0, h_0) + \frac{\partial G}{\partial h}(a_0, h_0) > 0; \quad (5)$$

$$\frac{\partial F}{\partial a}(a_0, h_0) \cdot \frac{\partial G}{\partial h}(a_0, h_0) - \frac{\partial F}{\partial h}(a_0, h_0) \cdot \frac{\partial G}{\partial a}(a_0, h_0) > 0; \quad (6)$$

$$\left(\frac{D_h}{D_a} \frac{\partial F}{\partial a}(a_0, h_0) + \frac{\partial G}{\partial h}(a_0, h_0) \right)^2 - \dots \quad (7)$$

$$\dots - \frac{4D_h}{D_a} \left(\frac{\partial F}{\partial a}(a_0, h_0) \cdot \frac{\partial G}{\partial h}(a_0, h_0) - \frac{\partial F}{\partial h}(a_0, h_0) \cdot \frac{\partial G}{\partial a}(a_0, h_0) \right) > 0,$$

where (a_0, h_0) is the (positive) homogeneous steady-state solution of:

$$F(a_0, h_0) = 0; \quad (8)$$

$$G(a_0, h_0) = 0. \quad (9)$$

The first two inequalities ensure that (a_0, h_0) is a linearly stable solution of Equations (2-3). The second two inequalities ensure the existence of modes unstable to spatial disturbances, thereby setting the conditions for the arising of asymmetric patterns.

One of the most widely used activation-inhibition model is named after Gierer and Meinhardt [17]:

$$\frac{\partial a}{\partial t} = \frac{\rho a^2}{h} - \mu_a a + D_a \nabla^2 a + \rho_a; \quad (10)$$

$$\frac{\partial h}{\partial t} = \rho a^2 - \mu_h h + D_h \nabla^2 h + \rho_h. \quad (11)$$

In such a model, a represents a short-range autocatalytic substance (activator) and h is its long-range antagonist, i.e., inhibitor. (This model can be regarded as a simplified version of another classical model, the so-called Brussellator [21].) Let us analyse it a bit more in detail.

We start with the left hand side of Equation (10). The term a tends to self-enhance (growth proportional to ρa^2), but such growth is slowed down by the inhibitor by a factor $\frac{1}{h}$. Further, the activator concentration decays proportionally to its value; this would correspond for example to the natural decay of molecules of type a to an inert state. The constant μ_a describes the rate at which such decay takes place. Molecules can move across nearby cells following the concentration gradient, hence the term $D_a \nabla^2 a$. The last term (ρ_a) is added in order to ensure that the process can initiate also in areas of low activator concentration.

As far as Equation (11) is concerned, the production of inhibitor molecules is fostered by the presence of activators according to a factor ρa^2 . The second term relates to natural decay, as described above, and the same applies to the other terms as well. A graphical representation of the model is reported in Fig. 1.

Figure 2 illustrates the typical pattern formation process resulting from this activator-inhibitor model: starting from a homogeneous mix of chemicals (left) that is slightly perturbed, the system progressively self-organizes into spot patterns (right), where the spots are regions of high activator concentration. Figure 3 shows the concentrations of activator and inhibitor at the end of the same simulation. Both figures 2 and 3 were produced using the parameters from [13], on a grid of size 32x32.

Such a model is simple enough to lend itself to a mathematical treatment, while at the same time representing a reasonable model for a variety of biological processes [17]. In order to get asymmetric patterns, it must have $D_h \gg D_a$ (the inhibitor diffuses much faster than the activator) and $\mu_h > \mu_a$ (the inhibitor drains more quickly than the activator). More formally, the conditions (4-7) have to be satisfied [17]. An example of parametrizing the Gierer-Meinhardt for a specific application will be discussed in Sec. 4.

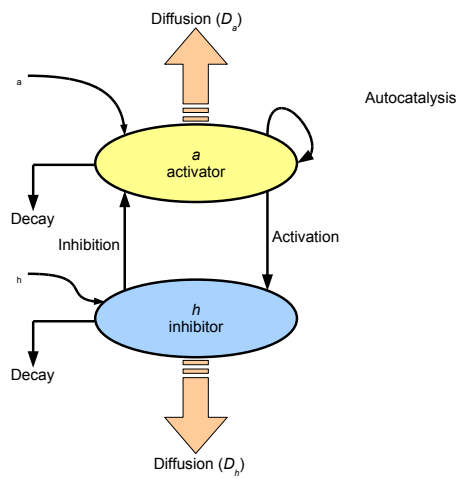


Fig. 1. Schematic representation of the Gierer-Meinhardt model.

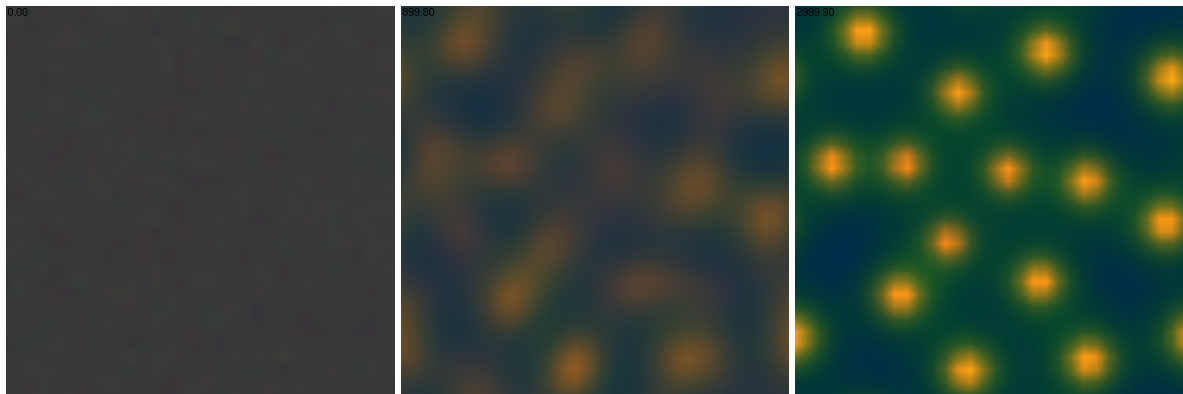


Fig. 2. Typical pattern formation resulting from the Gierer-Meinhardt model. Left: initial homogeneous mix of chemicals; middle: at 900 seconds of simulated time; right: after 3000 seconds, patterns are fully formed and stable.

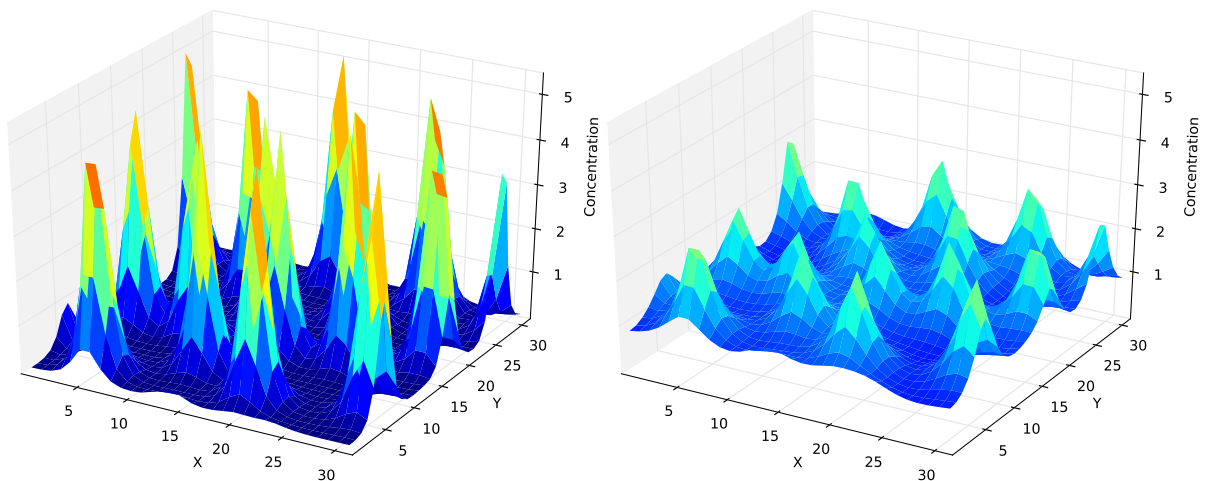


Fig. 3. Concentration levels of activator (left) and inhibitor (right) for the right-side pattern on Figure 2.

3.1 Activator-Depleted Substrate Model

Several variations of the basic activator-inhibitor model have been proposed in mathematical biology literature [7,17,18], some encompassing a larger number of equations, representing situations in which more complex interactions among molecules take place.

An interesting alternative approach in our context is the *activator-depleted substrate model* [17], or activator-substrate for short. Instead of modelling the explicit presence of a molecular species able to slow down the activation process, the activator-substrate model achieves a similar antagonistic effect through the depletion of a substrate s , which gets consumed during the production of the activator a . The resulting reaction-diffusion equations read:

$$\frac{\partial a}{\partial t} = \rho s a^2 - \mu_a a + D_a \nabla^2 a + \rho_a; \quad (12)$$

$$\frac{\partial s}{\partial t} = -\rho s a^2 - \mu_s s + D_s \nabla^2 s + \delta. \quad (13)$$

In this case, the substrate s is produced everywhere at constant rate δ . The substrate get consumed, during the production of activator a , at a rate ρ . The substrate also decays at rate μ_s . Such a model is basically equivalent to the basic Gierer-Meinhardt one, although it presents some distinctive features in terms of the resulting patterns, e.g., smoother peaks [17].

3.2 Equivalent Models in Discrete Space

All the models surveyed so far are based on the use of ordinary differential equations, assuming a continuum in both the time and space dimensions. While the continuous time assumption is not a critical one from an application perspective and an equivalent discrete-time model can be straightforwardly derived (albeit the effect of asynchronism among the updates at nodes may influence the resulting pattern [7]), the use of a discrete space model requires some in-depth analysis.

In terms of application to a distributed computing setting, indeed, the discrete nature of the space domain cannot be neglected. In most cases, we will be interested in evaluating an activation-inhibition-like dynamics at a well-defined restricted set of locations, corresponding, e.g., to the location of machines/nodes. The interconnection can be regular, giving rise to a grid-like organization, or irregular, resulting in a generic connectivity graph. In such cases, we need to find an equivalent of the models outlined above for a general discrete-space setting.

We start by considering a regular two-dimensional, four-neighbours grid with spacing h among nodes. In such a case, the Laplacian at node (i, j) can be replace, for a given variable x , by the following formula:

$$\nabla^2 x(i, j) = \frac{x(i+1, j) + x(i, j+1) + x(i-1, j) + x(i, j-1) - 4 \cdot x(i, j)}{h^2}. \quad (14)$$

The Laplacian operator can therefore be seen as a convolution with L , defined as:

$$L = \frac{1}{h^2} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (15)$$

Using a compact notation, for a general diffusion-reaction equation such as (1), the following formula results for the discrete-space case:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{f}(\mathbf{c}) + D \cdot L * \mathbf{c}. \quad (16)$$

Let us now briefly introduce the case in which nodes/machines (where reactions take place) are connected in a generic graph structure. We now denote by i the identity of a given node and by $\mathcal{N}(i)$ the set of its one-hop neighbours. We further consider the possibility that a link $i \rightarrow j$, $j \in \mathcal{N}(i)$ is

characterized by a “weight” factor $D_{i,j}$. (This would correspond to the diffusion coefficient along such an edge.) We can therefore express the discrete Laplacian operator at i as:

$$\nabla^2 x_i = \sum_{j \in \mathcal{N}(i)} D_{i,j} \cdot (x_j - x_i). \quad (17)$$

Additional details for the discrete time/discrete space case can be found in the cellular automata literature, and in particular in [7,22].

3.3 Cellular Neural Networks

Cellular neural networks (CNNs) represent a model for locally coupled identical dynamical systems (cells). Each cell can be regarded as a multiple-input, single-output, non-linear processing unit. Cellular neural networks have been introduced in [5] and they were initially meant to be used for processing signals, e.g., in image processing and/or pattern recognition. Later on, they found a variety of applications as model for distributed computation, whereby simple units (cells) can give rise to complex system-level behaviour [23].

The inputs from a cell come from neighbouring (locally-coupled) cells. At each time step the state of cell i is updated according to some non-linear function $F(\cdot)$, which takes as inputs the state of neighbouring cells and the current state x_i . The output is computed according to a function $G(x_i)$ and diffused to neighbouring cells, which in turn will use it as input for updating their own internal state. A schematic representation of CNN operations is reported in Figure 4.

CNNs are discrete models in both time and space domains. They can be used to emulate activation-inhibition patterns. For doing so, we need to define for each node i an activation domain \mathcal{D}_A^i and an inhibition domain \mathcal{D}_H^i . The evolution of the state of cell i can therefore be written as:

$$x_i(t+1) = f \left(\zeta x_i(t) + \sum_{j \in \mathcal{D}_A^i} \varepsilon_{i,j} x_j(t) + \sum_{j \in \mathcal{D}_H^i} v_{i,j} x_j(t) \right), \quad (18)$$

where ζ is the self-activation coefficient, $\varepsilon_{i,j} > 0$ is the activation coefficient for node i associated to node j and $v_{i,j} < 0$ is the inhibition coefficient for node i associated to node j and $f(\cdot)$ is a non-linear function³.

In case of a regular (grid) topology, CNNs can be regarded, from the computational standpoint, as equivalent to cellular automata.

4 Applications to Networking/Computing Problems

In this section we aim at providing an overview of applications of the models described in the previous section to problems in either networking or computing domains. In particular, we focus on the use of activation-inhibition mechanisms to the design of autonomous, distributed coordination schemes. It is worth remarking that we are not covering in this section application to procedural generation (a term used in computer graphics to refer to content generated algorithmically rather than manually), where reaction-diffusion mechanisms are widely used to generate random (but realistic) textures, landscapes, etc.

In [8] Durvy and Thiran proposed the use of activation-inhibition mechanisms to create transmission patterns in ad hoc wireless networks. Their aim is to devise efficient distributed mechanisms for regulating access to the channel by exploiting spatial reuse. In their model, each node has a medium access probability, which is evolved over time according to a cellular neural network model [5,23]. The authors prove analytically that, by appropriately dimensioning the CNN parameters, a spatial schedule respecting certain constraints on the distance among active nodes can be achieved. This corresponds to a schedule respecting certain non-interfering constraints among nodes, while at the same time achieving a good spatial reuse figure.

³ Typically $f(\cdot)$ is taken as the projection over a given interval.

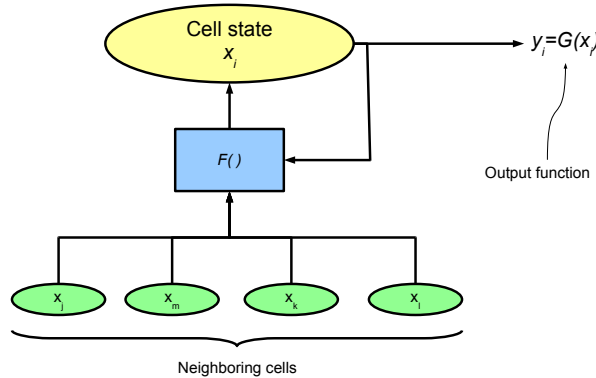


Fig. 4. Schematic representation of the Cellular Neural Network model.

The activation patterns achievable with reaction–diffusion systems can be also applied to congestion control problems in wireless networks. An approach has been proposed in [26] with application to wireless mesh networks. In this system, the reaction part changes the activation level of a node depending on the current buffer occupancy. The buffer size has a positive impact on the activation level. The diffusion part is used to limit channel contention among neighbouring units. Information on the current level of activator and inhibitor are communicated by means of broadcasting. In this way, nodes with a larger number of packets are given higher priority in accessing the channel, thereby effectively reducing congestion in the network.

Similar activation patterns can find application in clustering problems in wireless sensor networks. This problem has been addressed by Neglia and Reina in [19]. The authors devise a set of parameters for the Gierer–Meinhardt model, such that activation patterns respect a constraint on the distance among maxima in the activation level. They first retrieve conditions to ensure that non–trivial patterns can arise, according to conditions (4–7). In order to solve the system, they fix the stable operating point at $(a_0, h_0) = (1, 1)$ and get a set of conditions on the system parameters in equations (10–11). They then add additional conditions in order to ensure the presence of an unstable periodic mode (with period corresponding to the desired distance among activation maxima) and to ensure that all other modes will be stable. Simulation results confirm that their approach leads to the desired patterns, and moreover, show that it can outperform a probabilistic method whereby each node decides with a given probability (independently from its neighbours) to activate.

Activation patterns in wireless sensor networks were also considered in [11], where activation stripes are created to drive the movement of mobile robots (targeting mainly environmental monitoring applications). Their approach is based on the use of Thomas’ model, a substrate–inhibition model widely studied in mathematical biology [18]. They also propose the use of a pure diffusion model (without reactions) to create regular patterns (in particular: stripes).

The use of anisotropic diffusion in a CNN model was considered by Lowe and Miorandi to devise a distributed mechanism for building “data highways” in dense wireless sensor networks [15,14,16]. In their works, they aim at finding efficient routing structures for many–to–one (or many–to–some) communications patterns in wireless networks. Taking inspiration from some recent results on the capacity of wireless ad hoc networks [9] they aim at building a two–level hierarchical structure, whereby nodes communicating at minimum transmission power can construct connected trees routed in the data sink(s), called “data highways”. Such trees are used to drain data from the sources to sink(s); nodes not on the highways can connect to them by means of a single–hop high–power transmission. In order to reduce interference, a constraint on the distance among tree branches is given. Activator–inhibitor mechanisms are used to create spatial patterns meeting such constraints. In order to orient the highways towards the sink(s), anisotropic diffusion is considered, whereby either nodes are assumed to know the relative posi-

tion of the sink(s) [15] or can estimate it by using a gradient-based approach [14,16]. The width of the inhibition region is tuned in such a way to ensure a constraint on the distance among branches of the highways is guaranteed. As an example of the kind of patterns achieved, we report in Fig. 5 two graphs obtained for a network with 1000 nodes and one single sink (the parameters used to generate the network are the same reported in [16]):

- (a) Resulting activation field after 25 iterations of the anisotropic diffusion process;
- (b) Resulting structure (data highways) computed taking local maxima along the maximal gradient of the distance from the sink direction.

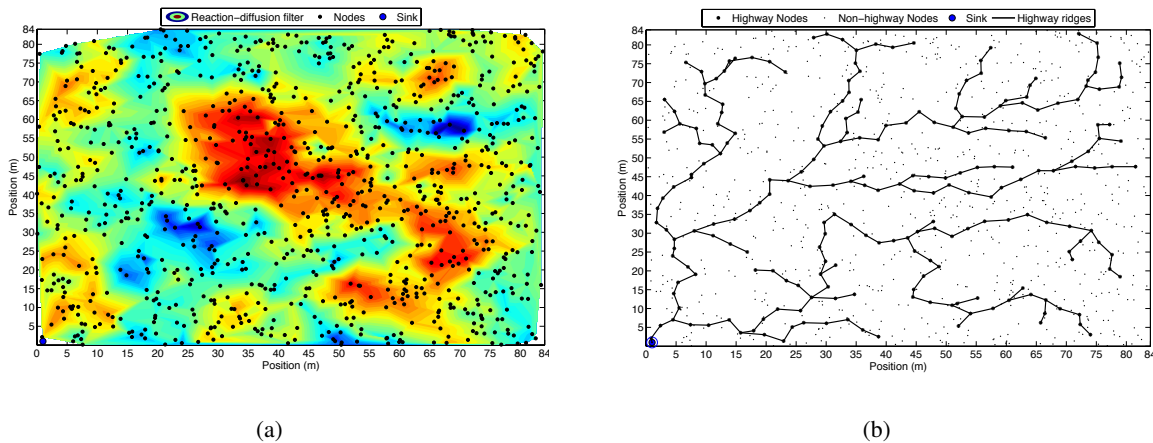


Fig. 5. Pattern arising using anisotropic diffusion in a wireless network with 1000 randomly placed nodes and one single sink: (a) Activation field after 25 iterations of the reaction-diffusion process; (b) Data highways resulting from tracing activation field ridges.

An interesting application of activation-inhibition patterns to the autonomous coordination of a distributed camera surveillance system is reported in [25]. In their system, each camera can adjust pan, tilt and zoom parameters. The overall objective is to decrease the area of blind spots in the whole surveillance area. By means of numerical simulations, the authors show that their system presents interesting robustness properties with respect to removal/rearrangement of cameras. A similar application is considered in [12], where the authors deal with a distributed camera surveillance system, where cameras are connected by means of wireless links. Cameras can adjust their video coding rate; the objective is to detect the presence of potential targets (e.g., intruders) and to provide high-resolution images of them. Areas that do not present potential targets can be transmitted at lower resolution. A simple reaction-diffusion model is used, and numerical simulations are performed to validate the ability of the proposed mechanism to effectively track the movement of targets and adjust accordingly the video coding rate.

5 Conclusions

In this chapter we presented a survey of activation-inhibition models and of their application to distributed coordination problems in networking and computing.

In general, we can conclude that activation-inhibition models can be profitably employed in situations in which the desired solution of the coordination problem can be mapped to a topological pattern that can be achieved by means of reaction-diffusion systems. However, using activation-inhibition mechanisms as a design tool is not an easy job. Models such as the Gierer-Meinhardt one are difficult to parametrize, and the resulting pattern may critically depend on the network structure and scale considered, which in many cases is not something that can be controlled on beforehand. Approaches based on cellular neural networks are in this respect more appealing, at least for cases in which regular, simple patterns shall be achieved.

One promising research direction that we believe is worth pursuing relates to the use of chemical computing languages for implementing, in a native way, activation–inhibition mechanisms. In this case, code fragments can be mapped to molecules, and the chemical reactions underpinning the model can be directly implemented in the programming language. In this way, activation–inhibition mechanisms can be used as a generic model for implementing network–based computations.

References

1. Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas Knight, Radhika Nagpal, Erik Rauch, Gerald Sussman, and Ron Weiss. Amorphous computing. *Communications of the ACM*, 43, May 2000.
2. Andrew Adamatzky, Benjamin De Lacy Costello, and Tetsuya Asai. *Reaction-Diffusion Computers*. Elsevier Science Inc., New York, NY, USA, 2005.
3. Yaneer Bar-Yam. *Dynamics of Complex Systems*. Westview Press, 2003.
4. M. C. Boerlijst and P. Hogeweg. Spiral wave structure in pre-biotic evolution: Hypercycles stable against parasites. *Physica D: Nonlinear Phenomena*, 48(1):17–28, 1991.
5. L. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Trans. on Circuits and Systems*, 35:1257–1272, 1988.
6. Andreas Deutsch and Sabine Dormann. *Cellular automaton modeling of biological pattern formation: characterization, applications, and analysis*. Birkhauser, 2005.
7. S. Dormann. *Pattern Formation in Cellular Automaton Models*. PhD thesis, University of Osnabrück, Dept. of Mathematics/Computer Science, 2000.
8. Mathilde Durvy and Patrick Thiran. Reaction-diffusion based transmission patterns for ad hoc networks. In *INFOCOM*, pages 2195–2205, 2005.
9. Massimo Franceschetti, Olivier Dousse, David N. C. Tse, and Patrick Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Transactions on Information Theory*, 53(3):1009–1018, 2007.
10. Boris Hasselblatt, Bernold Fiedler, and A. B. Katok. *Handbook of dynamical systems*, volume 2. Gulf Professional Publishing, 2002.
11. T. C. Henderson, R. Venkataraman, and G. Choikim. Reaction-diffusion patterns in smart sensor networks. In *Proc. of IEEE International Conference on Robotics and Automation*, volume 1, pages 654–658, 2004.
12. Katsuya Hyodo, Naoki Wakamiya, and Masayuki Murata. Reaction-diffusion based autonomous control of camera sensor networks. In *Proc. of BIONETICS*, Budapest, HU, 2007.
13. A. J. Koch and H. Meinhardt. Biological pattern formation: from basic mechanisms to complex structures. *Reviews of Modern Physics*, 66(4), October 1994.
14. D. Lowe and D. Miorandi. Reaction-diffusion generation of data highways in dense wireless sensor networks graphs. In *Proc. of SNA*, San Francisco, 2009.
15. David Lowe and Daniele Miorandi. All roads lead to rome: Data highways for dense wireless sensor networks. In *Proc. of S-Cube 2009: The first international conference on Sensor Systems and Software*. ICST, 2009.
16. David Lowe, Daniele Miorandi, and Karina Gomez. Activation–inhibition–based data highways for wireless sensor networks. In *Proc. of Bionetics 2009: 4th International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*. ICST, 2009.
17. H. Meinhardt. *Models of biological pattern formation*. Academic Press, London, UK, 1982.
18. James Dickson Murray. *Mathematical Biology: Spatial models and biomedical applications*. Springer, 2003. Volume 2 of *Mathematical Biology*.
19. Giovanni Neglia and Giuseppe Reina. Evaluating activator-inhibitor mechanisms for sensors coordination. In *Proc. of Bionetics*, Budapest, Hungary, 2007. ICST.
20. John E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, July 1993.
21. I. Prigogine and R. Lefever. Symmetry breaking instabilities in dissipative systems. *J. Chem. Phys.*, 48:1695–1700, 1968.
22. M. Sipper. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Springer-Verlag, Heidelberg, 1997.
23. P. Thiran. *Dynamics and self-organization of locally-coupled neural networks*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1997.
24. Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, B 327:37–72, 1952.
25. Atsushi Yoshida, Katsuji Aoki, and Shoichi Araki. Cooperative control based on reaction-diffusion equation for surveillance system. In *KES (3)*, pages 533–539, 2005.
26. Atsushi Yoshida, Takao Yamaguchi, Naoki Wakamiya, and Masayuki Murata. Proposal of a reaction-diffusion based congestion control method for wireless mesh networks. In *Proc. of IEEE ICACT*, Phoenix Park, Korea, 2008.

Branching Processes and their Generalization Applied to Wireless Networking

Eitan Altman¹ and Dieter Fiems²

¹ INRIA, BP93,

06902 Sophia Antipolis, France

Email: altman@sophia.inria.fr

² SMACS Research Group, Ghent University,

St-Pietersnieuwstraat 41, 9000 Gent, Belgium

Email: Dieter.Fiems@UGent.be

Abstract. In modeling large networks, one is often faced with complex systems involving large populations that evolve over time. To describe, model, analyze and optimize such systems, it is helpful to make use of paradigms from other branches of science. Perhaps more than any other discipline, Biology is relevant for that purpose. It has been the cradle of the theory of branching processes, of the mathematics of epidemics propagation, of dynamic population models of predator and prey, of evolutionary game theory, and other more. In this chapter we focus on branching processes, present some of the most recent advances in that area, and then describe a large number of applications to networking.

1 Introduction

The beginning of branching process theory is often attributed to Galton and Watson. In their time, the second half of the 19th century, there was a severe concern among aristocratic families that surnames were becoming extinct. The disappearance of a name of a family was considered as the death of the family and it was thought that the extinct families were replaced by families from lower social classes [2]. Francis Galton posed the question of computing the extinction probability of family names in the *Educational Times* of 1873 [28]. More precisely, assume that each man in generation n has some random number of sons in generation $n + 1$, according to a fixed probability distribution that does not vary from individual to individual. What is then the probability that a family dies out? The Reverend Henry William Watson replied with a solution [56]. Together, they then wrote a paper in 1874 entitled “On the probability of extinction of families” [29]. The solution of Watson was not complete though. He had come to the wrong conclusion that all families sooner or later die out (this behavior holds only in the “subcritical regime”).

Galton and Watson appear to have derived their process independently of the much earlier work by the French statistician I.J. Bienaymé [15] (1845), which was unknown till it was rediscovered in 1962 by Heyde and Senneta, see e.g. [39]. The correct answer to the problem of the extinction probability was answered by Bienaymé, but he did not publish a formal proof. This same problem seem to have been formulated independently by A.K. Erlang, and was published shortly after his death in [24]. The full proof of the problem was first given by Steffensen [55] who used an unpublished attempt of a solution by Erlang. Interestingly, Erlang is very well known by the networking community, independently of his contribution to branching processes: he is often seen as the one who started queueing theory and the mathematics of networking; the Erlang units used for dimensioning network capacity are named after him.

Applications of Galton-Watson processes in biology appeared already in the work of Fisher [26,27] and Haldane [35,36], who studied the rate with which rare mutations vanish from a population. Biologists not only made use of branching processes but also had an important impact on the theory of branching processes. The role of biology in the development of the mathematics of populations, and of branching processes in particular, can be learned from various books in that area, see e.g. [46] or from examples given in [48].

We briefly mention the impact of branching processes on queueing theory and on networking. Branching models are already used in the work of Borel in 1942 [17] and of Kendall in 1951 [41] for the busy period of an M/G/1 queue. Connections of branching processes with the processor sharing queue have been shown in 1988 in [57] and further exploited by Grishechkin in [31] (see the survey on processor sharing queues [58] for more recent related references). Recall that on the flow level, the processor sharing queue is, in turn, one of the mostly used models for throughput sharing over the Internet. In 1992, Grishechkin analyzed a retrial queue [32] and identified an underlying branching process with immigration. Also polling systems have been related to multi-type branching processes, see for example [52]. We have shown in [4] that the infinite server queue can also be modeled using branching processes with immigration. The queueing models above are quite useful for describing network protocols and models. In particular, polling systems have been used to model local area networks based on token passing rings. They have also been used in sensor ad-hoc networks [40], and in modeling bluetooth [61]. The infinite server queue has been used in [49] to derive the connectivity distance in ad-hoc networks on the line, e.g. for vehicular ad-hoc networks (VANETs). Branching processes have had some recent applications in networks with big populations. They have been used in P2P networks to model the propagation of the number of copies of a file [60]. Finally, delay tolerant networks have been successfully modeled and analyzed using branching processes in [25].

In this chapter we present the basic machinery of branching processes (Section 2) as well as some recent contributions extending the basic theory (Sections 4, 5 and 6). We illustrate the usefulness of these models through various examples in networking, including ferry based local area network (Section 3), the infinite server queue (Section 7) and delay tolerant networks (Section 8).

2 Galton-Watson Processes with Immigration

We introduce below the elementary Galton-Watson processes along with their extensions to continuous state space branching and multitype branching.

2.1 Integer Valued Scalar Branching Processes

We define the Galton-Watson branching process formally. Consider some population with an integer number of individuals. Let Y_n be the number of individuals in generation n . Starting with a fixed Y_0 , we recursively define

$$Y_{n+1} = \sum_{i=1}^{Y_n} \xi_n^{(i)}$$

where $\xi_n^{(i)}$ are i.i.d. random variables taking non-negative integer values. Defining

$$A_n(m) := \sum_{i=1}^m \xi_n^{(i)}, \quad (1)$$

we can rewrite the above as

$$Y_{n+1} = A_n(Y_n). \quad (2)$$

For the networking applications, the branching process is often combined with an additional component: an immigration process. Such a branching process with immigration is then defined through the recursion

$$Y_{n+1} = A_n(Y_n) + B_n. \quad (3)$$

The specific assumptions on B_n will be introduced later.

2.2 Continuous State Space Branching Processes

The definition above of branching processes has already been extended to a continuous state space in [37,38]. Various alternative equivalent definitions have appeared since then, see [1,14,42,43] and the

references therein. We recall our definition of [4]. We note that A_n is non-negative and has a divisibility property: for any non-negative integers m , m_1 and m_2 such that $m_1 + m_2 = m$, and for any n , we have

$$A_n(m) = A_n^{(1)}(m_1) + A_n^{(2)}(m_2)$$

where for each n , $A_n^{(1)}$ and $A_n^{(2)}$ are independent random processes, both with the same distribution as A_n . We take this property, together with the non-negativity of A_n as the basis to define the continuous state branching processes. Noting that these properties are satisfied by Lévy processes³, we define a continuous state branching process as one satisfying (2) where A_n is a non-negative Lévy process (a so-called subordinator).

2.3 Discrete Multitype Branching

Consider now d different types of individuals. An individual of type j at generation n may have offspring of several types. More precisely, the k th such individual will have a random number $\xi_{ji}^{(k)}(n)$ offspring of type i .

Consider the dynamics (3) where Y_n and B_n are now vectors. A_n is vector valued as well and is defined as follows. The i th element of the column vector $A_n(Y_n)$ is given by

$$[A_n(Y_n)]_i = \sum_{j=1}^d \sum_{k=1}^{Y_n^j} \xi_{ji}^{(k)}(n) \quad (4)$$

where $\xi^{(k)}(n)$, $k = 1, 2, 3, \dots$, $n = 1, 2, 3, \dots$ are i.i.d. random matrices of size $d \times d$. Each of their elements is a non-negative integer.

3 Ferry-based Wireless Local Area Network (FWLAN)

We identify the branching processes introduced above within a single application, a ferry based wireless local area network [40]. A number d of isolated nodes are scattered over some area. Communication between a node and the outer world or between nodes, is made possible via a message ferry. There is a fixed base station (BS) that is connected to the Internet (or to other base stations). The ferry brings all traffic from (respectively to) nodes in the FWLAN to (respectively from) the BS.

The ferry has a predetermined cyclic path which collects packets from a node and delivers packets to it when it is in the vicinity of the node. Call the BS node 0. The time to move from node i to node $i + 1$ (where we understand $i + 1$ to mean 0 when $i = d$) is distributed like a random variable $V(i)$.

We restrict ourselves to download traffic and assume that there is no traffic from nodes of the FWLAN to other nodes in the FWLAN. Assume that packets to node i arrive at the BS in accordance with a Poisson process with rate λ_i . The time to download a message has a general distribution with finite first and second order moments. Each time the ferry arrives at the base station, it (instantaneously) copies all the messages that have arrived since its last passage at the base station. Hence, a globally-gated polling system with $d + 1$ queues (queue i corresponds to node i , $i = 1, \dots, d$) can be used to model the FWLAN. The ferry (or server) cyclically moves between the different nodes (stations) which incurs walking times. When the ferry arrives at the BS (at station 0), the ‘‘global gate’’ of the polling system is closed. Only packets that were present then will be served in the following cycle.

Define the n th polling instant to be the n th time after $t = 0$ that the ferry arrives at the BS. Moreover, let the n th cycle be the time between the n th and $n + 1$ st polling instant. We now identify a number of branching processes.

³ A stochastic process $\{\phi = \{\phi_t, t \geq 0\}$ is said to be a Lévy process if (i) $\phi_0 = 0$ almost surely, (ii) For any $0 \leq t_1 < t_2 < \dots < \infty$, the increments $\{\phi_{t_{i+1}} - \phi_{t_i}, i = 1, 2, \dots\}$ are independent, (iii) For any $s > 0$, the stochastic processes $\{\phi_{t+s} - \phi_s, t \geq 0\}$ have the same distribution, (iv) ϕ_t is almost surely right continuous with left limits.

Example 1. Downlink model for a single station: the scalar case with discrete space. Consider the case $d = 1$, there is only a single node. Let Y_n denote the number of packets at the base station at the n th polling instant. Y_n then satisfies (3), with A_n defined in (1). $\xi_n^{(i)}$ in (1) corresponds to the number of arrivals during the download time of the i th packet after the n th polling instant. By the assumptions on arrivals and service times, this is an independent random variable. Moreover, B_n corresponds to the number of arrivals during the walking time (back and forth). If the walking times are stationary ergodic, so is B_n and we can compute the first two moments of Y_n (see further).

Example 2. Downlink model for several stations: scalar case with continuous space. Now, consider any number d of nodes. In spite of the multitype nature of the problem, the cycle time turns out to be a scalar branching process, albeit with a continuous state space. It satisfies,

$$C_{n+1} = \tilde{A}_n(C_n) + V_n.$$

Here $\tilde{A}_n(C_n)$ is the amount of “work” (the total download time) that arrives at the base station during the n th cycle. By the Poisson assumption and the independence of the service times, it is easily verified that $\{\tilde{A}_n\}$ is a sequence of i.i.d. Lévy processes. Furthermore, V_n is the total walking time during the cycle.

Example 3. Downlink model with several stations: multitype discrete branching process. Let Y_n denote the number of packets at the base station for the different nodes at the n th polling instant. In this case Y_n is vector valued and satisfies (3), with A_n defined in (4). $\xi_{ji}^{(k)}(n)$ in (4) equals the number of packets for the i th node that arrive during the download time of the k th packet at node j during the n th cycle. By the independence of the service times and the Poisson nature of the arrivals, it is seen that the independence assumptions on $\xi^{(k)}(n)$ hold. Finally, B_n is vector valued and its entries correspond to the number of arrivals for the different nodes during the total walking time in the n th cycle.

For all the examples above, it will be direct to compute the first two moments of the cycle times or of the vector Y_n in steady state using the theory introduced in Section 5. The first two moments are in fact what is needed to compute the expected waiting time in each node (the waiting time is defined as the time elapsed from the moment that a packet arrived at the BS till its download begins). Computations of these expected waiting times for a polling system that can be used to model our FWLAN are given in [19] for the case of independent vacations and in [3] for stationary ergodic walking times.

4 Extensions: Multitype Semi-Linear Processes

This section and the next one summarize extensions of branching processes and of the statistical framework in which they can be analyzed which we have developed in recent years (see [5] and the references therein).

Consider a column vector Y_n whose entries are Y_n^i , $i = 1, \dots, d$, where Y_n^i take values on the non-negative subset of \mathbb{R}^+ . Consider the following equation in vector form:

$$Y_{n+1} = A_n(Y_n) + B_n. \quad (5)$$

Our extensions with respect to branching processes that we have seen so far are the following.

- In the statistical assumptions made on the immigration process, we relax the standard assumption that B_n is i.i.d. (independent and identically distributed) or that it is a function of some Markov chain. We assume below that the d -dimensional column vector B_n is a stationary ergodic stochastic process whose entries B_n^i , $i = 1, \dots, d$, take values in the non-negative real numbers.
- Our framework extends continuous state branching processes to the multitype case. It further allows to handle hybrid models in which some state elements are discrete and other are real valued.
- In classical branching, the number of offspring of different individuals are independent. By relaxing this assumption we are able to handle branching processes and linear stochastic difference equations in a unified way.

For each n , we assume that A_n is a non-negative vector valued random field that is non-decreasing in its arguments. The sequence A_n is i.i.d. with respect to n , and $A_n(0) = 0$. We further assume that A_n satisfies the following conditions.

A1: $A_n(y)$ has the following **divisibility property**: if for some k , $y = y^0 + y^1 + \dots + y^k$ where y^m are vectors, then $A_n(y)$ can be represented as

$$A_n(y) = \sum_{i=0}^k \widehat{A}_n^{(i)}(y^i)$$

where $\{\widehat{A}_n^{(i)}\}_{i=0,1,2,\dots,k}$ are identically distributed with the same distribution as $A_n(\cdot)$. In particular, for any sequence $k(n)$, $\{\widehat{A}_n^{(k(n))}\}_n$ are independent.

Remark 1. For a given n , we do not assume that $\{\widehat{A}_n^{(i)}\}_{i=0,1,2,\dots}$ are independent.

A2: (i) There is some matrix \mathcal{A} such that for every y ,

$$\mathbb{E}[A_n(y)] = \mathcal{A}y.$$

(ii) The correlation matrix of $A_n(y)$ is linear in yy^T and in y . We shall represent it as

$$\mathbb{E}[A_n(y)A_n(y)^T] = F(yy^T) + \sum_{j=1}^d y_j \Gamma^{(j)}, \quad (6)$$

where F is a linear operator that maps $d \times d$ non-negative definite matrices to other $d \times d$ non-negative definite matrices and satisfies $F(0) = 0$.

Notice that when Y is random and independent of the process A_n , then assumption A2 yields the following properties,

$$\mathbb{E}[A_n(Y)A_n(Y)^T] = F(\mathbb{E}[YY^T]) + \sum_{j=1}^d \mathbb{E}[Y_j] \Gamma^{(j)}, \quad (7)$$

$$\text{cov}[A_n(Y)] = F(\text{cov}(Y)) + \text{cov}[A_n(\mathbb{E}[Y])]. \quad (8)$$

Finally, we need an additional assumption to assure the existence of a stationary regime. Let $\|\mathcal{A}\|$ denote the largest absolute value of the eigenvalues of \mathcal{A} . We make the following assumptions throughout the chapter.

A3: $\|\mathcal{A}\| < 1$ and $\mathbb{E}[B_0] < \infty$.

This condition not only ensures that a stationary regime exists, but also assures its uniqueness, and the convergence to this regime from any initial state (see [5] for details). We now consider some particular processes that adhere to these assumptions.

4.1 Linear Stochastic Differential Equations

As a first example, assume that $A_n(\cdot)$ is linear, and thus there is a random matrix, which we denote with some abuse of notation as A_n , such that $A_n(Y_n)$ can be written as $A_n Y_n$. In particular, in assumption A1, we have $\widehat{A}_n^{(i)} = A_n$. Now, A2 holds with

$$F(yy^T) = \mathbb{E}(A_n(y)(A_n(y))^T) = \mathbb{E}(A_n y y^T [A_n]^T).$$

We present some more insight on F . Let $(A_n)_i$ denote the i th row of A_n . Then

$$[A_n y y^T A_n^T]_{ij} = [A_n y]_i [A_n y]_j = (A_n)_i y [(A_n)_j y] = y^T ((A_n)_i)^T [(A_n)_j y]$$

Hence,

$$[F(y^T)]_{ij} = y^T \mathbb{E}[(A_n)_i]^T (A_n)_j y.$$

Stochastic difference equations where (A_n, B_n) can be general stationary ergodic sequences have been extensively studied. Stability conditions and asymptotic expressions for the stationary distributions can be found in [20,21,30] and references therein.

Linear stochastic difference equations have been used extensively to study the TCP congestion control protocol over the Internet. One and two dimensional models that describe the behavior of the protocol under a random loss process have been studied in [6,22,50]; a further random process describing the end-to-end delay is included in [7]. Models for any number of competing connections where losses are due to congestion have been studied in [8,13,54].

4.2 Discrete Multitype Branching Processes with Immigration

Consider the dynamics (3) where A_n is given by (4). We assume further that for any $l = 1, 2, 3, \dots, l' = 1, 2, 3, \dots, k = 1, \dots, d, i = 1, \dots, d, m = 1, \dots, d, j = 1, \dots, d$ and $m \neq k$, $\xi_{ki}^{(l)}(0)$ and $\xi_{mj}^{(l')}(0)$ are independent. Denote $\mathbb{E}[\xi_{ij}^{(k)}(n)] = \mathcal{A}_{ji}$. In this case, in assumption A1, $\widehat{A}_n^{(i)}$ are i.i.d. random variables.

Denote $\text{cov}(\xi)_{jk}^i = \mathbb{E}(\xi_{ij}^{(0)} \xi_{ik}^{(0)}) - \mathcal{A}_{ji} \mathcal{A}_{ki}$. Then, it is shown in [5]:

$$\mathbb{E}[(A_n(y))_i (A_n(y))_j] = \sum_{k=1}^d \sum_{m=1}^d \mathcal{A}_{ik} \mathcal{A}_{jm} y_k y_m + \sum_{k=1}^d y_k \text{cov}(\xi)_{ij}^k$$

Hence, $\Gamma^{(k)}$ is given by the matrix $\text{cov}(\xi)^k$ and assumption A2 holds.

Many processes in networks can be described by discrete multitype branching processes with immigration. We already obtained such a process for the ferry based wireless local network. Moreover, the dynamics of infinite server queues (see section 7) and of packet forwarding in delay tolerant networks (see section 8) can be described by means of these processes.

4.3 Continuous State Branching Processes with Immigration

As we had in the scalar case, continuous state multitype branching processes are also defined using Lévy processes. These are defined again through the recursion (3) but where all variables are now vector valued. In particular, A_n is now a non-negative additive Lévy field.⁴ For each A_n and for each $y \in \mathbb{R}_+^m$, $A_n(y)$ takes values in \mathbb{R}_+^d . We shall not go into details here on the definition of a non-negative additive Lévy field. The interested reader is referred to [5]. Instead, we mention that A2 holds indeed. In particular, $\text{cov}[A_n(y)]$ turns out to be linear in y , i.e. of the form $\sum_{j=1}^d y_j \Gamma^j$.

A continuous state branching process — albeit one-dimensional — was already identified for the Ferry based wireless local network (section 3). Further, the station times in polling systems with an exhaustive or a gated polling discipline can be described by this type of process [9].

5 First and Second Moments

We consider the general framework of Section 4 and now provide expressions for the first two moments. Denote by \bar{y}_i the first moment of the i th element of Y_0^* , the stationary regime, satisfying (5) and denote $\text{cov}(Y)_{ij} = \mathbb{E}[(Y_0^*)_i (Y_0^*)_j] - \bar{y}_i \bar{y}_j$. Let b_i and $b_i^{(2)}$ denote the first two moments of B_n^i and let b be a column vector with entries b_i . Define the following $d \times d$ matrices: $\mathcal{B}(k)$ is the matrix whose ij th entry equals $\mathbb{E}[B_0^i B_k^j]$, where k is an integer. \widehat{B} is the matrix whose ij th entry equals $b_i b_j$, and $\text{cov}(B)$ is the matrix whose ij th entry equals $\mathbb{E}[B_0^i B_0^j] - b_i b_j$. Finally, define $\widehat{\mathcal{B}}(k) := \mathcal{B}(k) - \widehat{B}$.

For ease of notation, let $F^n(\cdot) = F(F^{n-1}(\cdot))$, $F^1(\cdot) = F(\cdot)$, we then have the following theorem, see [5]⁵.

⁴ A random field is an extension of a stochastic process where the "time" parameter is not a scalar but a vector in \mathbb{R}_+^d .

⁵ Note that the formula for the covariance in [5] is not correct.

Theorem 1. (i) The first moment of Y_n^* is given by

$$\mathbb{E}[Y_0^*] = (I - \mathcal{A})^{-1}b. \quad (9)$$

(ii) Assume that the first and second moments b_i and $b_i^{(2)}$ are finite and that F satisfies

$$\lim_{n \rightarrow \infty} F^n = 0. \quad (10)$$

Then the matrix $\text{cov}(Y_0^*)$ is the unique solution of the set of linear equations:

$$\begin{aligned} \text{cov}(Y_0^*) = \text{cov}(B) + \sum_{r=1}^{\infty} \left(\mathcal{A}^r \widehat{\mathcal{B}}(r) + \left[\mathcal{A}^r \widehat{\mathcal{B}}(r) \right]^T \right) \\ + F(\text{cov}(Y_0^*)) + F(\mathbb{E}[Y_0^*] \mathbb{E}[Y_0^*]^T) - \mathcal{A} \mathbb{E}[Y_0^*] \mathbb{E}[Y_0^*]^T \mathcal{A}^T + \sum_{k=1}^d \bar{y}_k \Gamma^{(k)}. \end{aligned} \quad (11)$$

The second moment matrix $\mathbb{E}[Y_0^* (Y_0^*)^T]$ in steady state is the unique solution of the set of linear equations:

$$\mathbb{E}[Y_0^* (Y_0^*)^T] = \mathbb{E}[B_0 B_0^T] + \sum_{r=1}^{\infty} \left(\mathcal{A}^r B(r) + \left[\mathcal{A}^r B(r) \right]^T \right) + F(\mathbb{E}[Y_0^* (Y_0^*)^T]) + \sum_{k=1}^d \bar{y}_k \Gamma^{(k)}. \quad (12)$$

Remark 2. Note that the sums in (11) as well as in (12) are finite since the finiteness for all i of the second moments $b_i^{(2)}$ implies that $B(j)$ are uniformly bounded and since $\|\mathcal{A}\| < 1$. Note also that if for some i , $b_i^{(2)}$ is infinite, then it follows directly from (5) that $\mathbb{E}([Y_n]_i^2)$ is infinite for all $n > 0$ and thus also in the stationary regime.

Remark 3. We comment on the condition (10). It is equivalent to requesting that all eigenvalues of F have modulus smaller than one. To illustrate the necessity of this condition, consider the stochastic difference scalar equation

$$Y_{n+1} = A_n Y_n + B_n \text{ where } A_n = \begin{cases} 5 & \text{w.p. } 0.1, \\ 0.1 & \text{w.p. } 0.9. \end{cases}$$

A_n are assumed to be i.i.d. and assume $Y_0 = 0$. Then for all n

$$\mathbb{E}[Y_n] \leq \frac{b}{1 - 0.59}.$$

but

$$\mathbb{E}[Y_n^2] > b^{(2)} 2.5^n.$$

which diverges. Thus Y_n does not converge to a stationary ergodic regime (since $\mathcal{A} = 0.59 < 1$) but this limit has an infinite second order moment.

6 Examples of Correlated Immigration

We now focus on an example where the second moment can be calculated explicitly.

6.1 Markov modulated processes

We assume in this Subsection that B_n are random vectors whose distribution depends on an underlying ergodic Markov chain θ_n taking values in a finite space Θ . We denote the transition matrix of the Markov chain by \mathcal{P} and let $\pi(\theta)$ denote the unique steady state probability of being in state θ . Moreover, let g^i be the row vector with entries $\mathbb{E}[B_n^i | \theta_n = \theta]$, $\theta \in \Theta$, and let \hat{g}_i be the row vector with entries $\mathbb{E}[B_n^i | \theta_n = \theta] \pi(\theta)$. We then have the following lemma.

Lemma 1. In the Markov-correlated model described above, we have for $k > 0$,

$$[\mathcal{B}(k)]_{ij} = \mathbb{E}[B_0^i B_k^j] = \hat{g}^i \mathcal{P}^k [g^j]^T, \quad [\widehat{\mathcal{B}}(k)]_{ij} = \hat{g}^i \mathcal{P}^k [g^j]^T - b_i b_j. \quad (13)$$

Moreover, we have,

$$[\text{cov}(B)]_{ij} = \sum_{\theta \in \Theta} \pi(\theta) \mathbb{E}[B_0^i B_0^j | \theta_0 = \theta] - b_i b_j. \quad (14)$$

6.2 The Single Type Discrete Branching Process (One-dimensional Case)

We consider a scalar branching process, i.e. $d = 1$. Y_n in (5) is then a scalar instead of a vector and (4) simplifies to

$$A_n(Y_n) = \sum_{k=1}^{Y_n} \xi^{(k)}(n). \quad (15)$$

$\xi^{(k)}$ and \mathcal{A} are scalar too with $\mathbb{E}[\xi^{(k)}(n)] = \mathcal{A}$. Also note that $\|\mathcal{A}\| = |\mathcal{A}|$. Let $\mathcal{V} = \text{var}[\xi^{(k)}(n)]$, then Theorem 1 simplifies to:

Theorem 2. (i) The first moment of Y_0^* is given by

$$\mathbb{E}[Y_0^*] = \frac{b}{1 - \mathcal{A}}. \quad (16)$$

(ii) The variance of Y_0^* is given by

$$\text{var}[Y_0^*] = \mathbb{E}[(Y_0^*)^2] - (\mathbb{E}[Y_0^*])^2 = \frac{\text{var}[B] + 2 \sum_{r=1}^{\infty} \mathcal{A}^r \widehat{\mathcal{B}}(r) + \mathcal{V}(1 - \mathcal{A})^{-1}b}{1 - \mathcal{A}^2}.$$

Next, we further restrict the discussion to the Markovian setting of Section 6.1. This allows us to provide an explicit expression for $\sum_{r=1}^{\infty} \mathcal{A}^r \widehat{\mathcal{B}}(r)$.

Lemma 2. In the case of one dimensional state space with the Markov model for correlation, we have

$$\sum_{r=1}^{\infty} \mathcal{A}^r \widehat{\mathcal{B}}(r) = \hat{g} \mathcal{A} \mathcal{P} [I - \mathcal{A} \mathcal{P}]^{-1} g^T - \mathcal{A} (1 - \mathcal{A})^{-1} b^2,$$

and,

$$\text{var}[Y_0^*] = \frac{\text{var}[B] + 2\hat{g} \mathcal{A} \mathcal{P} [I - \mathcal{A} \mathcal{P}]^{-1} g^T - 2\mathcal{A} (1 - \mathcal{A})^{-1} b^2 + \mathcal{V}(1 - \mathcal{A})^{-1}b}{1 - \mathcal{A}^2}.$$

7 Infinite Server Queue

The infinite server queue has been frequently used in networking. Some examples are [49], [59], and the references therein. We here apply the general theory of the previous sections in order to compute the two first moments of the size of an infinite server queue in discrete time. Our model contains in particular the $G/PH/\infty$ queue; we have reported on that model already in [4]. This section summarizes results from [5].

A discrete time model of the infinite server queue, similar to the one we present here, has been studied independently in [23] with great generality. The derivation there does not rely on the branching process framework, and is achieved using a one dimensional discrete time state model. The fact that we base our analysis on a multidimensional branching process approach allows us to obtain stronger results: (i) we are able to analyze (in Section 7.5) a network of $G/GI/\infty$ queues. Moreover, (ii) our framework allows us to obtain correlations between the number of customers in different phases for a discrete $G/PH/\infty$ queue.

7.1 A Discrete Branching Model

Service times: Service times are considered to be i.i.d. and independent of the arrival process. We represent the service time associated to any customer in the queue as the discrete time analogy of a phase type distribution: there are d possible service phases. The initial service phase of a customer is chosen at random according to some probability $p(k)$. If at the beginning of slot n a customer is in a service phase i then it will move at the end of the slot to a service phase j with probability P_{ij} . With probability $1 - \sum_{j=1}^d P_{ij}$ it ends service and leaves the system at the end of the time slot.

Modeling the service time Let $\xi^{(k)}(n), k = 1, 2, 3, \dots, n = 1, 2, 3, \dots$ be i.i.d. random matrices of size $d \times d$. Each of its element can take values of 0 or 1, and there is at most a single 1 on each row. The ij th element of $\xi^{(k)}(n)$ has the interpretation of the indicator that equals one if at time n , the k th customer among those present at service phase i moves to phase j . Obviously, $\mathbb{E}[\xi_{ij}^{(k)}(n)] = P_{ij}$. P is a sub-stochastic matrix (it has non-negative elements and its largest eigenvalue is strictly smaller than 1), which means that services ends in finite time w.p. 1 and that $(I - P)$ is invertible.

Arrivals: Let $B_n = (B_n^1, \dots, B_n^d)^T$ be a column vector for each integer n , where B_n^i is the number of arrivals at the n th time slot that start their service at phase i . B_n is assumed to be a stationary ergodic sequence and to have finite expectation.

The recursive equation: Let Y_n^i denote the number of customers in phase i at time n . Then Y_n satisfies the recursion (5) where A_n is given by (4). In particular, $\mathcal{A} = P$ and indeed we have $\|\mathcal{A}\| < 1$ so that assumption A3 holds. We can therefore apply the results of the previous sections to get the first two moments.

7.2 Main Results

Corollary 1. *Theorem 1 holds for the $G/PH/\infty$ queue. Moreover, the first and second moment of the number of customers at the system in stationary regime are given by $\mathbf{1}^T(I - \mathcal{A})^{-1}b$ and $\mathbf{1}^T \text{cov}(Y_0^*)\mathbf{1}$, respectively, where $\mathbf{1}$ is a column vector with all entries 1's.*

Remark 4. We present a simple interpretation of the expression of the first moment of the number of customers at the system. Denote by λ the expected number of arrivals per slot. Clearly $\lambda = |b|$ where $|b|$ is the sum of the entries of the vector b . Define ζ to be the expected service time of an arbitrary customer and let $\rho = \lambda\zeta$. We shall first compute ζ . The ij th element of the matrix $(I - \mathcal{A})^{-1}$ has the interpretation of the total expected number of slots that a customer that had arrived at service phase j spent in phase i . Thus the j th entry of the vector $\mathbf{1}^T(I - \mathcal{A})^{-1}$ has the interpretation of the total expected number of slots that a customer arriving in service phase j spent in the system. Let $\beta = b/|b|$ be a vector whose entries equal the fractions of the arrivals that arrive in the different phases, then,

$$\zeta = \mathbf{1}^T(I - \mathcal{A})^{-1}\beta,$$

and

$$\rho = (\mathbf{1}^T(I - \mathcal{A})^{-1}\beta)|b| = \mathbf{1}^T(I - \mathcal{A})^{-1}b,$$

which is our expression for the first moment of the number of customers at the system. This relation is known to hold in fact for general $G/G/\infty$ queues, see e.g. [12, p. 134].

7.3 The Case of Geometric Service Times

We now study the special case of geometrically distributed service times. In other words, if at the beginning of slot n a customer is in the system then it will end service at the end of the slot with some probability α (it thus remains in the system during a geometrically distributed duration). Recall that the dimension d is determined by the number of service phases; here we have a single phase in which we remain with probability α . Thus the problem can indeed be formulated using random variables (dimension one) instead of random vectors. Y_n is a scalar and denotes the number of customers in the system. $\xi_n^{(k)}$ has the interpretation of the indicator that the k th customer present at the beginning of time-slot n will still be there at the end of the time-slot. Thus the probability that a customer in the system finishes its service within a time slot is precisely $\alpha = 1 - \mathcal{A}$. Moreover, since $\xi_n^{(k)}$ is Bernoulli distributed, we also have, $\mathcal{V} = \mathcal{A}(1 - \mathcal{A})$. We directly apply the results of Section 6.2 below.

To illustrate the one dimensional case obtained with service times that are geometrically distributed, we consider the following simple scenario. The arrival process depends on a Markov chain as in Subsection 6, and moreover, there can be either one or no arrival at a time slot.

We consider a Markov chain with two states $\{\gamma, \delta\}$ with transition probabilities given by

$$P = \begin{pmatrix} 1 - \varepsilon p & \varepsilon p \\ \varepsilon q & 1 - \varepsilon q \end{pmatrix}.$$

$\varepsilon > 0$ is a parameter that will be varied later in order to vary the correlations. The steady state probabilities of this Markov chain are

$$\pi = \left(\frac{q}{p+q}, \frac{p}{p+q} \right).$$

Hence

$$b = \mathbb{E}[B] = \mathbb{E}[B^2] = \frac{qp\gamma + pp\delta}{p+q}, \quad (17)$$

where $p_\gamma := P(B_n = 1 | \theta_n = \gamma)$, and $p_\delta := P(B_n = 1 | \theta_n = \delta)$. Equation (17) then implies the following:

$$\text{var}[B] = \frac{(qp\gamma + pp\delta)(q(1-p_\gamma) + p(1-p_\delta))}{(p+q)^2} \quad (18)$$

Note that $\pi, b, \mathbb{E}[B^2]$ and $\text{var}[B]$ do not depend on ε .

Applying the first part of Theorem 2 we get the following expression for the expected number of customers in the system in stationary regime:

$$\mathbb{E}[Y_0^*] = \frac{1}{1-\mathcal{A}} \frac{qp\gamma + pp\delta}{p+q}. \quad (19)$$

Some algebra then further yields the following expression for $\text{var}[Y^*]$ [5]:

$$\begin{aligned} \text{var}[Y_0^*] = & \frac{1}{(1-\mathcal{A}^2)(p+q)^2} \left((qp\gamma + pp\delta)(q(1-p_\gamma) + p(1-p_\delta)) \right. \\ & \left. + \frac{2\mathcal{A}pq(p_\gamma - p_\delta)^2(1-\varepsilon(p+q))}{1-\mathcal{A}+\varepsilon(p+q)\mathcal{A}} + \mathcal{A}b(p+q)^2 \right). \quad (20) \end{aligned}$$

7.4 A Numerical Example

As a numerical example for the model introduced in the last section, we set the following parameters: $p = q = 1$, $p_\gamma = 1$, $p_\delta = 0.5$. Substituting these parameters in (20), we obtain the following expression:

$$\text{var}[Y_0^*] = \frac{1}{(1-\mathcal{A}^2)} \left(\frac{3}{16} + \frac{\mathcal{A}(1-2\varepsilon)}{8(1-\mathcal{A}+2\varepsilon\mathcal{A})} + \frac{3}{4}\mathcal{A} \right). \quad (21)$$

In Fig. 1 we plot the variance of the steady state number of customers, $\text{var}[Y_0^*]$, while varying ε and \mathcal{A} .

Recall that for a fixed \mathcal{A} , the expectation of Y_0^* does not depend on ε . The variance of Y_0^* on the other hand is seen to be quite sensitive to the correlation between the B_n 's as determined by the parameter ε . This sensitivity is seen to increase as \mathcal{A} increases and the sensitivity is largest when \mathcal{A} approaches 1. We see that $\varepsilon = 1$ gives the smallest value of $\text{var}[Y_0^*]$ and that $\text{var}[Y_0^*]$ increases as ε decreases. For $\mathcal{A} = 0.7$ we get a difference of around 32% between the lowest and the largest value of ε , where as for $\mathcal{A} = 0.9$ we obtain a difference of 47%.

7.5 Extension to a Network

P2P networks in which a file is decomposed into K chunks have been modeled in [18] using a network of K infinite server queues in series. This motivates us to consider an arbitrary network of M stations, each with an infinite number of servers. The service time at station i has a set \mathcal{N}_i of d_i phases. Let $d = d_1 + \dots + d_M$. For any $j = 1, \dots, d$, let $s(j)$ denote the station to which j corresponds, i.e. if $j \in \mathcal{N}_i$ then $s(j) = i$.

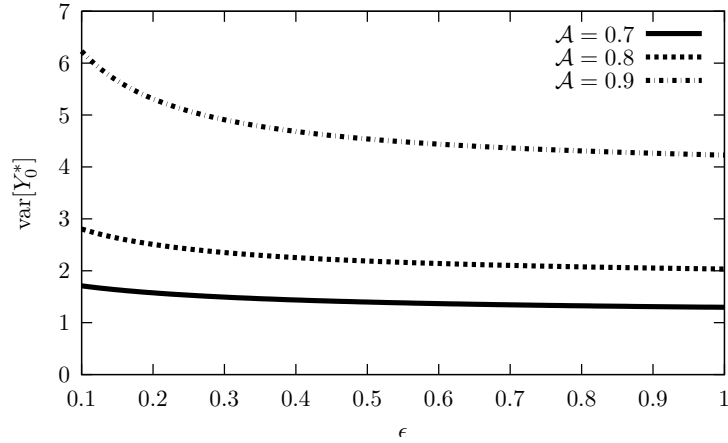


Fig. 1. $\text{var}[Y_0^*]$ (vertical axis) as a function of ϵ (horizontal axis) and for different values of \mathcal{A}

If at time n a customer is at phase j in station $s(j)$ then it either moves to another phase at the same station or moves to another phase in another station; the next phase k (either at the same station or at another one) is chosen with probability P_{jk} ; with probability $1 - \sum_{k=1}^d P_{jk}$ the customer leaves the system. Again we assume that the choices of the next phases are independent.

Let $B_n = (B_n^1, \dots, B_n^d)^T$ be a column vector for each integer n , where B_n^i is the number of arrivals at the n th time slot that start their service at phase i in station $s(i)$. B_n is assumed to be a stationary ergodic sequence.

With this description we see that we can identify the whole network as a single server station problem with an infinite number of servers and with d phases. Thus we can apply all the previous results.

8 Delay-Tolerant Mobile Ad Hoc Networks

Consider a sparse network that consists of a variable number of mobile nodes and one fixed source node. Time is discrete and it is assumed that at each time n , each node has a probability $p > 0$ to meet each other node. The validity of (a continuous time version of) this model without the random environment has been discussed in [34], and its accuracy has been shown for a number of mobility models (Random Walker, Random Direction, Random Waypoint).

We assume that the mobile nodes do not cooperate, and they receive the file only when they meet the source. The objective of having copies of the file in the network is to be able to transmit them to some potential customers. Moreover, it is assumed that each of the nodes (except for the source node) stays in the system during a geometrically distributed time with some parameter q . The source never leaves the system. Finally, at slot n , B_n mobiles that do not have the file join the system.

Let $\xi_n^{(i)}$ denote the indicator that equals 1 if the i th node without the file meets the source node and receives the file. Moreover let $\zeta_n^{(i)}$ ($\widehat{\zeta}_n^{(i)}$) denote the indicator that equals 1 if the i th node with (without) the file remains in the system at time n . Let X_n and Y_n denote the number of nodes with and without the file, respectively. We then obtain the following recursion,

$$X_{n+1} = \sum_{i=1}^{X_n} \zeta_n^{(i)} + \sum_{i=1}^{Y_n} \widehat{\zeta}_n^{(i)} \xi_n^{(i)},$$

$$Y_{n+1} = \sum_{i=1}^{Y_n} \widehat{\zeta}_n^{(i)} (1 - \xi_n^{(i)}) + B_n.$$

In vector notation this set of equations can be written as follows,

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = A_n \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + B_n \times \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

with,

$$A_n \begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^x \zeta_n^{(i)} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \sum_{i=1}^y \widehat{\zeta}_n^{(i)} \begin{pmatrix} \xi_n^{(i)} \\ 1 - \xi_n^{(i)} \end{pmatrix}.$$

Given the assumptions on residence times and connection probabilities, this shows that our framework is applicable. New nodes can arrive in accordance with a stationary ergodic process. The results of section 5 then yield the first and second moment of the number of mobile nodes that have and do not have a copy of the file.

9 Conclusions

We have presented some classical as well as novel tools in multitype branching processes (and their extensions) with immigration, that enable us to compute the first two moments of the state process in stationary regime. Interestingly, explicit expressions are obtained in spite of the weak (non-Markovian) assumptions on the immigration process. The applicability of branching process theory in the performance evaluation of networks is demonstrated by the identification of branching processes that capture the dynamics of various network elements.

References

1. S.R. Adke and V.G. Gadag, "A new class of branching processes", *Branching Processes: Proceedings of the First World Congress*, C.C.Heyde (Editor), 1-13, Springer Lecture Notes 99, 1995.
2. K. Albertsen, The Extinction of Families, *International Statistical Review / Revue Internationale de Statistique*, Vol. 63, No. 2 (Aug., 1995), pp. 234-239
3. E. Altman, "Stochastic recursive equations with applications to queues with dependent vacations," *Annals of Operations Research* 112(1):43–61, 2002.
4. E. Altman, "On stochastic recursive equations and infinite server queues," *Proceedings of IEEE Infocom*, Miami, March 13-17, 2005.
5. E. Altman, "Semi-linear stochastic difference equations," *Discrete Event Dynamic Systems* 19:115–136, 2009.
6. E. Altman, K. Avrachenkov and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses," *IEEE/ACM Transactions on Networking* 13(2):356–369, 2005.
7. E. Altman, C. Barakat and V.M. Ramos Ramos, "Analysis of AIMD protocols over paths with variable delay," *Computer Networks* 48(6):960–971, 2005.
8. E. Altman, D. Barman, B. Tuffin and M. Vojnovic, "Parallel TCP Sockets: Simple Model, Throughput and Validation," *Proceedings of IEEE Infocom*, Barcelona, 2006.
9. E. Altman and D. Fiems, "Expected waiting time in symmetric polling systems with correlated vacations," *Queueing Systems* 56:241–253, 2007.
10. K.B. Athreya, A.N. Vidyashankar, "Branching Processes," in *Handbook of statistics 19: Stochastic Processes: Theory and Methods*, Edited by D.N. Shanbhag, Elsevier, 2001.
11. K.B. Athreya and P. Jagers (Eds.), *Classical and Modern Branching Processes Series: The IMA Volumes in Mathematics and its Applications*, Vol. 84, Springer Berlin Heidelberg New York, 1997.
12. F. Baccelli and P. Brémaud, *Elements of Queueing Theory*, Springer, second edition, 2003.
13. F. Baccelli and D. Hong, "AIMD, fairness and fractal scaling of TCP traffic," *Proceedings of IEEE Infocom*, New York, 2002.
14. J. Bertoin. *Lévy Processes*. Cambridge University Press, 2002.
15. I.J. Bienaymé, "De la loi de la multiplication et de la durée des familles," *Soc. Philomath.*, Paris Extraits Ser. 5, 37–39, 1845.
16. J.D. Biggins, H. Cohn and O. Nerman, "Multi-type branching in varying environment." *Stochastic Processes and their Applications* 83:357–400, 1999.
17. E. Borel, "Sur l'emploi du théorème de Bernoulli pour faciliter le calcul d'une infinité de coefficients. Application au problème de l'attente à un Guichet," *Comptes Rendus Hebd. des Séanc. de l'Académie des Sciences* 214:452–456, 1942.
18. D.S. Menasche, A.A. Aragao Rocha, E. de Souza e Silva, R.M. Meri Leao, D. Towsley and A. Venkataramani, "Modeling Chunk Availability in P2P Swarming Systems," *The Eleventh Workshop on Mathematical Performance Modeling and Analysis*, 2009.
19. O. Boxma, H. Levy and U. Yechiali, "Cyclic reservation schemes for efficient operation of multiple-queue single-server systems," *Annals of Operations Research* 35:187–208, 1992.
20. A. Brandt, P. Franken and B. Lisek, *Stationary Stochastic Models*, Akademie-Verlag, Berlin, 1992.
21. A. Brandt. The stochastic equation $y_{n+1} = a_n y_n + b_n$ with stationary coefficients. *Advances in Applied Probability* 18:211–220, 1986.
22. V. Dumas, F. Guillemin and P. Robert, "A Markovian analysis of additive-increase multiplicative-decrease algorithms", *Advances in Applied Probability* 34(1):85–111, 2002.

23. I. Eliazar, "The discrete-time $G/GI/\infty$ queue," *Probability in the Engineering and Informational Sciences* 22:557-585, 2008.
24. A.K. Erlang, Opgave Nr. 15, *Matematisk Tidsskrift B*. pp. 36, 1929.
25. D. Fiems and E. Altman, "Markov-modulated stochastic recursive equations with applications to delay-tolerant networks," INRIA Research Report No. 6872, a shorter version has appeared in *Bionetics*, Avignon, 2009.
26. R.A. Fisher, On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, vol 42, pp 321-341, 1922.
27. R.A. Fisher, The distribution of generations for rate mutations. *Proceedings of the Royal Society of Edinburgh*, vol 50, 204-219, 1930
28. F. Galton, Problem 4001. *Educational Times* April 1 (17), 1873.
29. F. Galton and H.W. Watson, "On the probability of the extinction of the families", *J. Royal Antropol. Soc.*, London, vol. 4, pp. 138-144, 1874.
30. P. Glasserman and D.D. Yao, "Stochastic vector difference equations with stationary coefficients," *Journal of Applied Probability*, Vol 32, pp 851-866, 1995.
31. S.A. Grishenchkin, "On a relation between processor sharing queues and Crump-Mode-Jager branching processes", *Advances in Applied Probability*, **24**, 653-698, 1992.
32. S.A. Grishenchkin, "Multiclass batch arrival retrial queue analyzed as branching processes with immigration", *Queueing Systems* 11, 395-418, 1992.
33. R. Groenevelt and E. Altman, "Analysis of alternating-priority queueing models with (cross) correlated switchover times", *Queueing Systems*, Vol. 51, pp. 199-247, 2005.
34. R. Groenevelt, P. Nain and G. Koole, "Message Delay in Mobile Ad Hoc Networks" Proc. of PERFORMANCE 2005, Juan-les-Pins, France, October 3-7, 2005. *Performance Evaluation*, Vol. 62, No. 1-4, pp. 210-228, October 2005.
35. J.B.S. Haldane, A mathematical theory of natural and artificial selection, part V: Selection and mutation. *Proceedings of the Cambridge Philosophical Society*, vol 23, pp 838-844, 1927.
36. J.B.S. Haldane, The equilibrium between mutation and random extinction, *Annals of Eugenics*, Vol 9, 400-405, 1939.
37. M. Jirina, "Stochastic branching processes with continuous state-space", *Czechoslov. Math. J.* 8 (2) (1958), pp. 292-313.
38. M. Jirina, "Stochastic branching processes with a continuous space of states", *Theory Probab. Appl.* 4 (4) (1959), pp. 482-484.
39. C.C. Heyde and E. Senneta, *Studies in the History of Probability and Statistics*. XXXI. The simple branching process, a turning point test and a fundamental inequality: A historical note on I. J. Bienaymé, *Biometrika* 1972 59(3):680-683, 1972.
40. V. Kavitha and E. Altman, "Queueing in Space: design of Message Ferry Routes in static adhoc networks", 21st International Teletraffic Congress (ITC 21) September 15-17, 2009, Paris, France
41. D.G. Kendall, "Some Problems in the Theory of Queues", *J. Roy. Statist. Soc.*, Ser. B, vol. 13, pp. 151-185. 1951.
42. A. Lambert. The genealogy of continuous-state branching processes with immigration. *Journal of Probability Theory and Related Fields*, 122(1):42-70, 2002.
43. J.F. Le Gall, *Random trees and spatial branching processes*, Maphysto Lecture Notes Series (Univ of Aarhus), vol 9, 2000.
44. E. Key. Limiting distributions and regeneration times for multitype branching processes with immigration in a random environment. *Annals of Probability*, 15:344-353, 1987.
45. D. Khoshnevisan, Y. Xiao, and Y. Zhong. Local times of additive Lévy processes. *Stochastic Processes and their Applications*, 104:193-216, 2003.
46. M. Kimmel and D.E. Axelrod, *Branching Processes in Biology*. Springer, Series: Interdisciplinary Applied Mathematics , Vol. 19 2002.
47. A.N. Kolmogorov, "On the solution of a biological problem", *Proc. of Tomsk University*. Vol 2, 7-12, in Russian, 1938.
48. P. Haccou, P. Jagers, V.A. Vatutin, "Branching processes: variation, growth, and extinction of populations", Cambridge University Press, 2005.
49. D. Miorandi and E. Altman, "Connectivity in Ad-Hoc Networks: a Queueing Theoretical Approach", *Wireless Networks*, 2006.
50. N. Moller, C. Barakat, K. Avrachenkov and E. Altman, "Inter-protocol fairness between TCP New Reno and TCP Westwood+", in: Proceedings of NGI 2007 (Conference on Next Generation Internet Networks), Trondheim, Norway, May 2007.
51. M.P. Quine, "The multi-type Galton-Watson process with immigration", *J. Appl. Prob.* 7, pp 411-422, 1970.
52. J.A.C. Resing, "Polling systems and multi-type branching processes," *Queueing Systems* 13 (1993) 409-426.
53. B.A. Sevastyanov, "Limit theorem for branching processes of special form", *TPA* 2, pp. 121-136, in Russian, 1957.
54. R. Shorten, F. Wirth and D. Leith, "A positive systems model of TCP-like congestion control: asymptotic results", *IEEE/ACM Transactions on Networking*, Volume 14, Issue 3, pp. 616 - 629, 2006.
55. J.G. Steffensen, "Deux problemes du calcul des probabilités", *Ann. inat. H. Poincare*, vol. 3, pp. 331-344, 1933.
56. H.W. Watson, Solution to problem 4001. *Educational Times* August 1, 115-116, 1873.
57. S.F. Yashkov, The Non-Stationary Distribution of Numbers of Calls in the M/G/1 Processor-Sharing Queue, Proc. 3rd Int. Symp. on Systems Analysis and Simulation, Berlin: Akademie, 1988, vol. 2, reprinted in *Advances in Simulation*, P.A. Lukar and B. Schmidt, Eds., Berlin: Springer, vol. 2, pp. 158-162, 1988.
58. S.F. Yashkov, A.S. Yashkova, "Processor Sharing: A Survey of the Mathematical Theory" *Automation and Remote Control*. Vol 68, No 9, pp 1662-1731, 2007.
59. M. Zukerman, "Bandwidth allocation for bursty isochronous traffic in a hybrid switching system", *IEEE Trans. on Comm.*, **37**(12), Dec. 1989.
60. X. Yang, G. de Veciana, "Performance of Peer-to-Peer Networks: Service Capacity and Role of Resource Sharing Policies", *Performance Evaluation*, Volume 63, Issue 3, March 2006, Pages 175-194.
61. G. Zussman, A. Segall, and U. Yechiali, On the Analysis of the Bluetooth Time Division Duplex Mechanism, *IEEE Trans. on Wireless Communications*, Vol. 6, No. 6, pp. 2149-2161, June 2007.

On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security

Paolo Dini¹, Daniel Schreckling²

¹ Department of Media and Communications
London School of Economics and Political Science
London, United Kingdom
p.dini@lse.ac.uk

² Institute of IT-Security and Security Law
University of Passau
Passau, Germany
ds@sec.uni-passau.de

Abstract. This paper begins to chart and critically analyse the formal connections between algebra, logic, and cell biology on the one hand, and algebra, logic, and software security on the other. Much of the discussion is necessarily conceptual. Where the discussion is more formal the current distance between these disciplines appears evident. The paper focuses on the algebra of network coding, reviews the main types of algebraic and temporal logics that underpin security, and briefly discusses recent work in the application of algebra and logic to the DNA code.

1 Introduction

In this article we wish to begin to discuss some ideas related to the relevance of biology to security. In the BIONETS context, the application of biological concepts and models to security is complicated by the fact that the communication and computation framework of the BIONETS architecture, services, and protocols is supposed to be autonomic, i.e. adaptive, self-optimising, self-healing, etc. One of the fundamental assumptions of the project is that the desired autonomic behaviour can best be achieved through reliance on biologically-inspired communication and computation models. Additionally, the BIONETS disconnected network architecture implies a high level of local autonomy. As a consequence, the autonomic properties of the system need to be developed on top of a distributed P2P architecture and a dynamic network topology. Because biological systems are able to construct order and useful behaviour through bottom-up emergent and decentralised processes, we see that there is good alignment between the BIONETS architecture and the fundamental assumption of strong reliance on biology. In other words, rather than seeing the disconnected, decentralised, and P2P architecture as an additional requirement or in fact *burden*, biology seems to tell us that such an architecture is in itself one of the *enablers* of the desired self-organising and autonomic behaviour. Because reasonably secure centralised systems over IP networks can already be developed, clearly the challenge is to achieve the required security characteristics in a distributed and disconnected P2P environment whilst retaining the autonomic behaviour. For this reason it seems sensible to assume that also security properties can and should be achieved through biologically-inspired models.

In addition to the above considerations, our design philosophy calls for the integration of the security architecture with the system architecture from the very beginning of the research and design effort. Therefore, any fundamental rethinking of networking and computing principles that may be necessary to achieve autonomic behaviour of BIONETS networks must necessarily be integrated with the theoretical and architectural principles of security. Where the latter are found lacking or inadequate, new ideas must be developed hand-in-hand with the rest of the system. It is with this methodological requirement of theoretical integration in mind that the research discussed in this paper was performed.

The work reported in this chapter is meant to be complementary to evolutionary computing. In the Technical Annex we contrast the two main concepts in biology: evolution and self-organisation. By the

latter we mean all the processes relating to the life of the individual organism, thus a better name could be 'development', or 'morphogenesis', or 'gene expression'. Thus, in this paper we emphasise development over evolution. As Stuart Kauffman says,

Darwin's answer to the sources of the order we see all around us is overwhelmingly an appeal to a single singular force: natural selection. It is this single-force view which I believe to be inadequate, for it fails to notice, fails to stress, fails to incorporate the possibility that simple and complex systems exhibit order spontaneously. [1]

Spontaneous construction of order in biology happens at different time scales. The slowest process is through evolution by natural selection, so in evolution order is constructed across many generations, in response to a selection pressure. During the life of the individual morphogenesis relates to the growth period from the embryo to the adult. The processes that run the metabolism of the individual take place on the order of days, hours, or minutes. Finally, our mental processes, which are far from understood, take place on the order of microseconds to seconds. The table below summarises these facts for a species whose individuals live for a few decades.

Process	Mechanism	Result	Time Scale
Evolution	genetic operators natural selection	phylogenetic trees	millions of years
Morphogenesis	gene expression cell differentiation	adult organism	years
Metabolism	gene expression	structure and behaviour	days-hours- minutes
Cognition	neuronal network	thought	milliseconds

Thus, if evolution is the model that is able to explain phylogeny (a succession of organic forms sequentially generated by reproductive relationships), in this paper we are addressing the construction of a model that may eventually be relevant to ontogeny (the history of structural changes in a particular living being) ([2], cited in [3]). The former relates to the first row of the table, while the latter to the second and third rows. Before we can begin to understand and model morphogenesis, however, we need to understand gene expression, on which morphogenesis depends. The biology part of this paper is therefore focused entirely on the cell. Our objective is in fact to develop a model inspired by cell metabolic processes that can represent equally well biological and computing processes. The motivation for studying gene expression over evolution is that it is a much faster and more powerful process of self-organisation.

As discussed in Section 4 and in [4], although we can make the mapping from cell metabolic cycles to digital algorithms seem plausible, we still face the problem of the absence of physical interaction forces between digital entities, and of the concept of temperature. In other words, we cannot rely on the minimisation of free energy as the driver of software systems. More importantly, the interaction forces bring with them a built-in regularity that is a direct consequence of the regularity and universality of physical laws and that, it seems plausible to conclude, gives rise to the observed regularities in structure and behaviour of biological systems. In other words, not only do bio-chemical systems approach equilibrium spontaneously due to these interaction forces, but the manner in which they do so is constrained by the characteristics of physical law and by the spatio-geometrical properties of the constituent elements of matter to behave in certain ways and not others. For example, the six-fold symmetry of the snowflake and its predominantly flat shape are a consequence of the minimum energy configuration provided by 7 water molecules, which happens to be a co-planar configuration (this can be easily demonstrated by six identical coins surrounding another identical coin at their centre). Having accounted for all the actors that participate in the spontaneous construction of order of biological systems, by elimination the culprit must be the interplay between physical law and the spatio-geometrical characteristics of matter, in the presence of an energy flow through an open non-equilibrium system.

Our research is based on the assumption that the regularities that result from this interplay can be formalised through the mathematical theory of groups, which is a branch of abstract algebra. Parallel work by [5] supports this perspective. If we then identify the flow of energy with a flow of information we do not really need to worry about the lack of interaction forces. The behaviour of the users of the

software and communication system will provide a constant flow of information which, in our view, can be constrained by algebraic transformation and interaction rules to produce ordered structures and behaviour. In reference to Fig. 1, we can therefore see why there is an arrow between algebra and cell biology.

This point is quite important and should be stated again in different words. After 4 years spent researching the problem of realising biologically-inspired computing, as part of the DBE project [4] [3] [6] [7], the first author arrived at the following rationale or argument, which can be taken as a starting position for the research discussed in this article:

- The self-organisation exhibited by biological systems is driven by interaction forces and entropy maximisation (minimisation of free energy).
- The order, symmetry, and regularities exhibited by biological systems, furthermore, are a consequence of the regularities and symmetries in the underlying physical laws.
- In the absence of interaction forces or of the concept of temperature in digital systems, Kauffman’s view of self-organisation (which is complementary to Evolution) may only be realisable in digital systems through the imposition of artificial constraints that embody a structure analogous to that caused by physical interaction laws in biological systems.
- Over the past 3000 years, abstract algebra developed largely as an effort to formalise in the most general way possible the regularities that we perceive in the world around us. It therefore seems like a good starting point for the task at hand.
- The importance and effectiveness of abstract algebra, symmetries, fields, and groups is demonstrated well by network coding and erasure coding techniques, themselves being pursued as part of the BIONETS research.

After we began working on this task, we realised that, in fact, logic is also strongly related to algebra. This has strengthened our conviction that the direction we are working in is very interesting for the agenda of developing an effective theory of biologically-inspired computing.

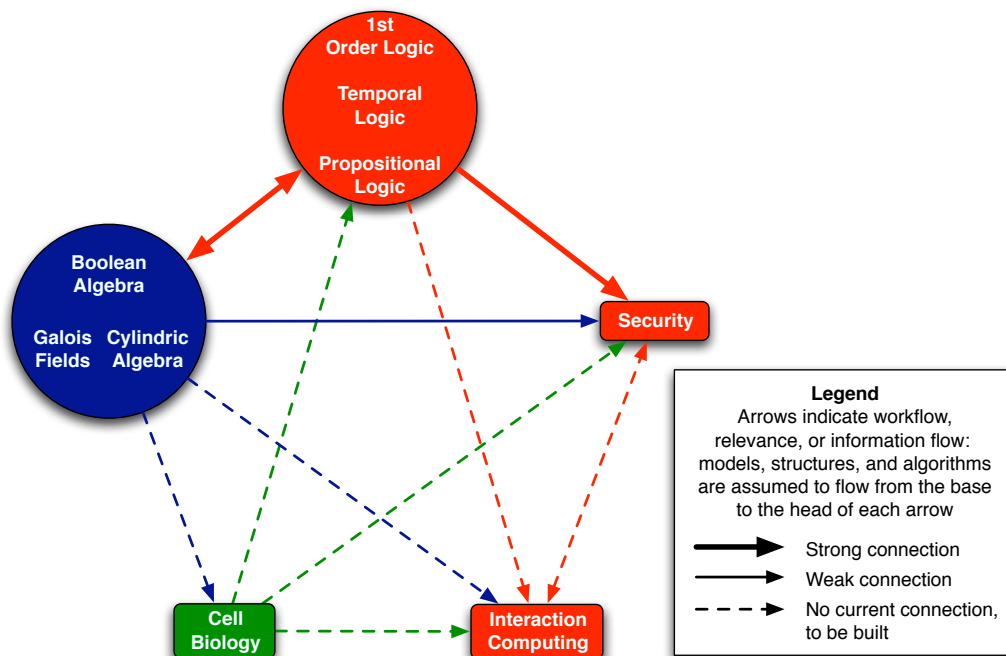


Fig. 1. Disciplinary connections of relevance to bio-inspired security

The arrow from cell biology to Interaction Computing is more difficult to explain, partly because the concept of interaction computing is still being formed [6] [7]. In order for the scenario described above

to function, the digital system needs to become *reactive* to the inputs and the behaviour of the users. In other words, there cannot be computation without interaction. Iterating this concept recursively to components that are farther removed from the user interface, they too cannot change their states without being 'pushed' by the components that precede them. The picture that emerges can therefore be characterised conceptually as a set of coupled and interacting finite state machines whose state spaces are subdivided into permissible regions bounded by surfaces defined by algebraic structure laws. Each state machine is performing a piece of the algorithm. This is thus what we mean by interaction computing for the execution of a distributed algorithm that is partly resident in the 'DNA' of the digital system and partly in the environment.

The ultimate objective of this paper is to show how a connection from biology to security might be possible through algebra and logic within a framework of interaction computing. In 1847 George Boole [8] published a short book that represented the first attempt to express logic rules and transformations through algebra. The immense impact that this small book had on logic and eventually computer science and electronics is well known to all. In this article we wish to examine this first bridge and relate it to the structures of abstract algebra. Such a connection has been investigated thoroughly by many authors since the publication of Boole's book. The algebraisation of logic has been one of the success stories of the 20th Century [9] [10] [11]. Therefore, we do not expect to say anything fundamentally new in this regard, in this article. However, if we take the interface between algebra and logic to be the 'centre of gravity' of this article, we are interested in shedding some light in two different directions. On the one hand, we will review the mapping of propositional logic to Boolean algebra, briefly explain the algebraisation of First-Order Logic, and finally show the link between First-Order and Temporal Logic, which is widely used in the security domain to support applications that are able to describe, detect, and prevent specific security features or breaches, respectively. On the other hand, we will review the fundamentals of abstract algebra, touching on coding and Galois theory, in order to explore its connections to the interaction between the DNA code and cell metabolism. In this manner we hope to begin charting a possible path from order construction in biology to autonomic security algorithms and data structures that weaves its way through algebra and logic.

Section 2 provides a very elementary introduction to abstract algebra concepts. Section 3, which has appeared in [12], applies the abstract algebra concepts to this familiar area of networking. Section 4 presents a discussion of basic logic concepts and their algebraic character, and shows their relevance to security. Finally, Section 5 draws some conclusions.

2 Abstract Algebra

In this section we will present the very basics of abstract algebra, which is relevant to coding theory and Galois theory. The objective is two-fold, i.e. to show how the structure of algebra resembles to some extent the structure of logic, to be presented in the next chapter, and to provide a vocabulary and formalism that will at a later date be applied to biology. In this short article we will not attempt to review and define an isomorphic map between the parts of algebra and logic that are specifically relevant to biology and security, mainly because this is still an open question. We will also not attempt to "solve" any cell biology problems. The algebra discussed in this section is quite elementary because we think it is useful to have self-contained discussion of the concepts alongside the review of logic and algebraic logic presented in Section 3 of this article. Our purpose is to take the first tentative steps in the direction of an integrative formalism that we hope will bear fruits over the life of the BIONETS project.

2.1 Starting Concepts

The language of abstract algebra is set theory. Since it is easy to read and understand for most people, it is used here. Abstract algebra deals with the operations between the elements of a set, with the mappings between different sets, and with mappings defined from a set to itself. Different kinds of operations and mappings give rise to different relationships between these constitutive parts of algebra, which is partly responsible for associating a *structure* with algebra. The elements of a set can be anything at all, including other sets. They can be integers, polynomials, functions, 'variables', and so forth. What

Boole realised is that they could also be the constitutive elements of logic, which provides one of the fundamental motivations for the line of investigation pursued in this article. Finally, we are making an explicit assignment of some yet unspecified elements of the cell as elements of a set. The temptation is to treat the cell itself as a set. Although this is not necessarily wrong, we are starting with much simpler sets such as, for example, treating the four bases of the DNA as the elements of a set that we can call the alphabet of DNA. Another example could be the 20 amino-acids from which all proteins are built.

Algebra has successfully identified and formalised the fact that certain sets, although apparently different on the surface, exhibit identical behaviour in how their elements relate to or combine with each other. This motivates the definition of abstract 'objects' that model the structure and/or behaviour of sets that recur again and again in countless applications. We will now define these basic building blocks, relying almost exclusively on examples from various sets of numbers (the integers \mathbb{Z} , the real numbers \mathbb{R} , etc.) and of polynomials. The following statements can be found in any algebra book, but we are borrowing freely mainly from [13] [14] [15].

A **binary relation** is a statement which, for any two elements of a set, is either true or false for that pair. Another way to put it is that a binary relation R on a set A is a subset of the Cartesian product $A \times A$. For example, if $A = \{1, 2, 3\}$, then the relation 'less than' on A is the set $\{(1, 2), (1, 3), (2, 3)\}$.

Let R be a binary relation on A . An **equivalence relation** is a binary relation that satisfies these three conditions:

- (Reflexive) $(a, a) \in R, \forall a \in A$
- (Symmetric) If $(a, b) \in R$, then $(b, a) \in R$
- (Transitive) If $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$

The **equivalence class** of the element $a \in A$ is the set

$$\{b \in A : (a, b) \in R\}$$

In the example above, the relation 'less than' fails all three properties, so it is not an equivalence relation. An important fact that is not necessarily obvious is that the set of equivalence classes of a relation R on a set A is a **partition** of A . In other words, the equivalence classes do not intersect and, together, they cover all of A .

2.2 Groups, Rings and Fields

Before we can move forward we need to define three of the abstract objects, or types of set, that were indicated above: groups, rings and fields.

A **group** is a set G of elements with a binary operation between them \circ such that:

- (G0- Closure) $\forall g, h \in G, g \circ h \in G$
- (G1- Associative) $g \circ (h \circ k) = (g \circ h) \circ k, \forall (g, h, k) \in G$
- (G2- Identity) $\exists e \in G: g \circ e = e \circ g = g, \forall g \in G$
- (G3- Inverse) $\forall g \in G, \exists h \in G: g \circ h = h \circ g = e$
- (G4- Commutative) $g \circ h = h \circ g, \forall (g, h) \in G$

The last condition is satisfied only by so-called abelian or commutative groups.

A **ring** R is a set with two operations, that are usually called 'addition' and 'multiplication', such that:

- (A0- Closure) $\forall a, b \in R, a + b \in R$
- (A1- Associative) $a + (b + c) = (a + b) + c, \forall (a, b, c) \in R$
- (A2- Zero) $\exists 0 \in R: a + 0 = 0 + a = a, \forall a \in R$
- (A3- Inverse) $\forall a \in R, \exists b \in R: a + b = b + a = 0$
- (A4- Commutative) $a + b = b + a, \forall (a, b) \in R$
- (M0- Closure) $\forall a, b \in R, ab \in R$
- (M1- Associative) $a(bc) = (ab)c, \forall (a, b, c) \in R$

- (D- Distributive) $(a + b)c = ac + bc$, $c(a + b) = ca + cb \forall (a, b, c) \in R$

Rings can have additional structure if they satisfy one or more of these additional axioms:

- (M2- Identity) $\exists 1 \in R (1 \neq 0): a1 = 1a = a, \forall a \in R$
- (M3- Inverse) $\forall a \in R (a \neq 0), \exists b \in R: ab = ba = 1$
- (M4- Commutative) $ab = ba, \forall (a, b) \in R$

A ring that satisfies M2 is called a **ring with identity**. If it satisfies M2 and M3 it is called a **division ring**. A ring that satisfies the first 8 axioms plus M4 is a **commutative ring**. A ring that satisfies *all* the axioms, i.e. a commutative division ring with identity, is a **field**. Subrings and subfields are subsets of rings or fields that are themselves rings or fields.

In preparation for the next section on logic, it is interesting to represent the field $\{0, 1\}$ using the language of truth tables, as shown in Table 1.

		Operations	
		+	·
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 1. Truth table for the \mathbb{Z}_2 field

2.3 Cosets and Homomorphisms

Now we are going to start building up some more complicated concepts using the basics defined so far. It is probably fair to say that the previous section is an essential reference that we can go back to when the derivations below get confusing, whereas Section 2.1 will be used right away.

The presentation of this material is necessarily abstract and a little dry. We will try to compensate for this by spending extra effort in explaining the concepts thoroughly, whenever possible, since most of the concepts are extremely interesting. Given also their great depth, however, in most cases we will not be able to do justice to the beauty and broad relevance of the theory. We will use numbers and polynomials as motivating examples, keeping an eye out for connections with logic, which is presented in the next chapter, and with biology, which is presented in the last chapter.

A particular kind of equivalence class that we are going to need is the coset. To define a coset we need a ring R and one of its subrings, S . What we say is that we partition R into the cosets of S . The 'co' in 'coset' stands for 'complementary'; therefore, S itself is *one of* the cosets. For this to work out we can't just take a random subset of R and call it S . The fact that the cosets partition (are disjoint and cover) R tells us that each coset could be an equivalence class. In fact, this is precisely how they are defined: a **coset** of S (or S itself) is the equivalence class of a relation E on the ring R that satisfies the rule

$$(a, b) \in E \text{ if } b - a \in S, \quad a, b \in R \quad (2.1)$$

Written in this way, this definition seems to imply that S must be known in advance. In fact, we just need to know a characteristic of S that allows us to construct it by 'filtering' the elements of R .

For example, take as a coset the equivalence class of the binary relation 'equality mod 5, with remainder 0' on the ring $R = \mathbb{Z}$. An algorithm to construct this coset involves picking two integers a and b and asking: does $a \bmod 5 = b \bmod 5 = 0$? If yes, a and b belong to S , which is clearly the infinite set

$$S = \{\dots, -15, -10, -5, 0, 5, 10, 15, \dots\}. \quad (2.2)$$

We notice that the difference between any two members of this set is indeed another member of this set, but we did not use this fact to construct S in this example. This shows how we already know something quite fundamental about S before constructing it, which is an example of what we mean by algebraic structure.

If we look at all the other equivalence classes that we could build from the binary relation 'mod 5' we will construct an important object, \mathbb{Z}_5 . Allenby [14] defines it in general as follows:

\mathbb{Z}_n is the set of equivalence classes determined by the equivalence relation ' $= \pmod n$ '.

So it is a set of sets, each of which looks similar to Eq. (2.2). Because this is hard to write down, we define a new concept, the **coset representative**, as an element of a coset that is convenient to use because it makes it easy to recognise what coset it refers to. For example, the most convenient coset representative of Eq. (2.2) is 0. The set of sets \mathbb{Z}_5 , therefore, can be conveniently written down as

$$\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}, \tag{2.3}$$

Each of these numbers 'points' to an infinite subset of \mathbb{Z} . These subsets of \mathbb{Z} are the 5 cosets referred to as \mathbb{Z}_5 , and they partition \mathbb{Z} . A useful shorthand to refer to each coset is to notice that S can be identified with the 0 coset representative, and can be related to the others as follows:

$$\begin{aligned} S &= \{\dots, -15, -10, -5, 0, 5, 10, 15, \dots\} \rightarrow 0 \\ S + 1 &= \{\dots, -14, -9, -4, 1, 6, 11, 16, \dots\} \rightarrow 1 \\ S + 2 &= \{\dots, -13, -8, -3, 2, 7, 12, 17, \dots\} \rightarrow 2 \\ S + 3 &= \{\dots, -12, -7, -2, 3, 8, 13, 18, \dots\} \rightarrow 3 \\ S + 4 &= \{\dots, -11, -6, -1, 4, 9, 14, 19, \dots\} \rightarrow 4 \end{aligned} \tag{2.4}$$

After all this work, which has enabled us to arrive at a notation that we will use again later, we can recognise this strange set of sets \mathbb{Z}_n as simply the set of remainders when \mathbb{Z} is divided by a particular integer n . This particular 'mechanism' will be an essential intuitive and mnemonic device when we start working with polynomials.

We can now gear up to define homomorphisms. First, Fig. 2 provides a quick graphical reminder of the basic types of mappings. Homomorphisms are mappings that can be defined between algebraic objects of the same type. We will focus on rings. A **ring homomorphism** is a map

$$\theta: \langle R, +, \cdot \rangle \rightarrow \langle S, \oplus, \odot \rangle \tag{2.5}$$

between rings that satisfies the following conditions for any $a, b \in R$:

$$\theta(a + b) = \theta(a) \oplus \theta(b) \tag{2.6a}$$

$$\theta(a \cdot b) = \theta(a) \odot \theta(b) \tag{2.6b}$$

This is a fairly abstract set of conditions since the elements of rings, as we said, can be anything and addition or multiplication can likewise represent more general binary operations between the elements of a ring than their familiar arithmetical interpretations. A homomorphism does not have to be 1-1 and onto. If it is 1-1 and onto, then it is an **isomorphism**. To gain an intuitive understanding on how constraining an homomorphism is, we are going to use an example that is actually an isomorphism, simply because it is familiar and easier to grasp. Then we will try to generalise a bit.

Example. If the map is from \mathbb{Z} to itself and is given by

$$y(x) = \frac{1}{2}x,$$

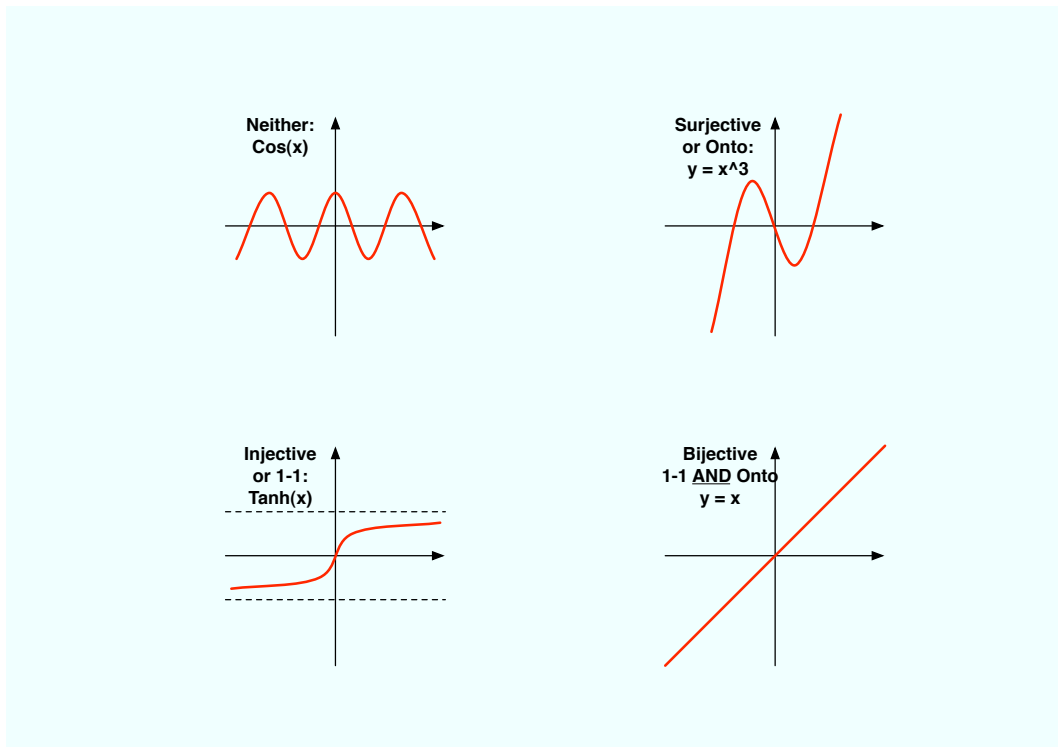


Fig. 2. The four basic kinds of mappings

then we can see that

$$y(2 + 4) = 3 = 1 + 2 = y(2) + y(4)$$

$$y(2 \cdot 4) = 4 \neq 1 \cdot 2 = y(2) \cdot y(4).$$

Remarkably, the simple straight line equation $y = (1/2)x$ is not a homomorphism (and therefore it is not an isomorphism either)! To get a homomorphism we need to use $y(x) = x$. Because $y(x) = x$ is 1-1 and onto, it is also an isomorphism. In fact, an isomorphism from a ring to itself is called an **automorphism**, which in less technical terminology can be called a **symmetry**.

2.4 Kernel, Image, Ideals and Factor Rings

This example has shown us that even if homomorphisms don't need to be 1-1 or onto, they are still fairly restrictive maps. Later we will have to come back to this discussion with an example that deals with somewhat more complicated elements than the integers. For now we can proceed with a couple of Venn diagrams, taken from Allenby.

Figure 3 shows two new objects, the kernel and the image of a homomorphism. The **kernel** of a ring homomorphism $\theta: R \rightarrow S$ is a subring of R defined by

$$\text{Ker}(\theta) = \{k \in R: \theta(k) = 0\}. \quad (2.7)$$

The **image** is a subring of S defined by

$$\text{Im}(\theta) = \{s \in S: s = \theta(r) \text{ for some } r \in R\}. \quad (2.8)$$

There is a generalisation of the kernel that we are going to need. An **ideal** of a ring R is a subring S of R such that, for any $s \in S$ and $r \in R$, $rs, sr \in S$. To decide whether a subring is an ideal we should test this

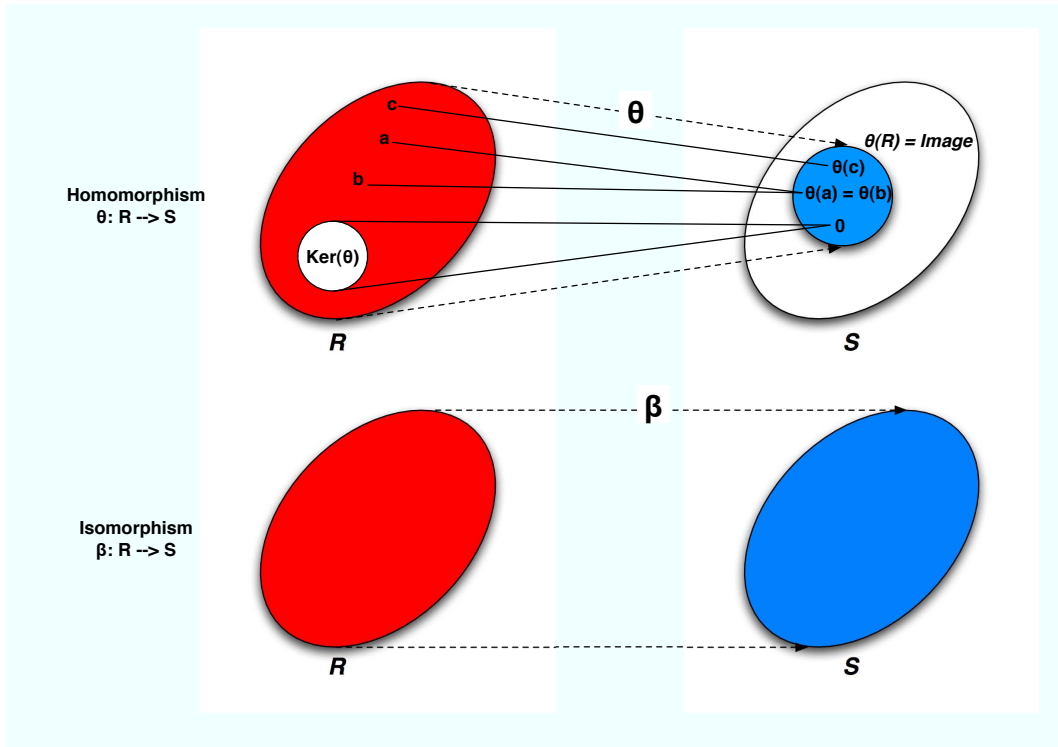


Fig. 3. Showing how homomorphisms are not 1-1 and Onto

condition plus all the ring axioms. However, there is an equivalent and shorter 'ideal test'. A non-empty subset S of a ring R is an ideal of R if and only if

$$(a) \forall s_1, s_2 \in S, s_1 - s_2 \in S \tag{2.9a}$$

$$(b) \forall s \in S \text{ and } r \in R, rs, sr \in S. \tag{2.9b}$$

On the basis of this we can say that the kernel of a homomorphism $\theta: R \rightarrow S$ is an ideal of R . For example, all the subrings of \mathbb{Z} of the form $n\mathbb{Z}$ are ideals. We can now see how the first of the sets in Eq. (2.4) is actually an ideal, which means that we can refer to the various cosets as $I, I + 1, I + 2$, etc. The 'set of sets' that we keep referring to is so important that it is given a special name, the factor ring. Additionally, its definition is made quite general, as follows.

Let I be an ideal of the ring R . The **factor ring** R/I is the set of cosets of I in R , with operations of addition and multiplication defined as

$$(I + x) + (I + y) = I + (x + y) \tag{2.10a}$$

$$(I + x)(I + y) = I + xy \tag{2.10b}$$

2.5 Fields

The research discussed in this article is not in mathematics, but in the application of algebra to logic, biology, and new forms and models of computing. Therefore, practically no proofs are given and a lot is taken for granted. We are just covering the smallest possible number of essential concepts in order to provide a basis for coding theory and Galois theory that can expose the underlying algebraic machinery. We are assuming that this algebraic machinery will then be relevant to biology, logic, and security. Furthermore, we hope that it will help us develop a model for interactive computing that should enable the replication of biological behaviour in software and in security applications in particular. Fields are the next algebraic structure that will help us in this endeavour.

Whereas the 'ring archetype' is the set of integers \mathbb{Z} , the 'field archetype' is the set of rational numbers, \mathbb{Q} . We saw in Section 2.2 that fields are more constrained than rings. In fact they are so constrained

that there are only two methods to construct a field. The first method essentially replicates the construction of \mathbb{Q} from \mathbb{Z} . We proceed with a few definitions taken from [13].

A **zero divisor** in a ring R is a non-zero element $a \in R: \exists (b \neq 0) \in R$ with $ab = 0$. An **integral domain** is a commutative ring with identity that has no zero-divisors (e.g. \mathbb{Z}). Let R be an integral domain. A field F is a 'field of fractions' of R if

- (a) R is a subring of F
- (b) Any element of F can be written in the form ab^{-1} for some $a, b \in R$

The result of this kind of construction is an infinite field.

The second method generalises the construction of the integers modulo n from the integers, that is, the field is a factor ring R/I . Some conditions need to be satisfied for this to be true, but in the examples we will discuss it will be true, so we skip the finer points here. The result of this kind of construction is a finite field.

Something we have not yet stated is that, for factor rings generated as a set of integers modulo n , n must be a prime number. More generally, Galois proved that any and all finite fields have order a prime power. This means that the number of elements of a finite field can **only** be either a prime (exponent = 1) or a prime raised to a positive integer power. For example, finite fields of size 5, or 25, or 256 exist, whereas no finite fields of size 18, or 30, or 100 exist. It is important to realise that, if the field is a prime power where the power is greater than 1, then such a field cannot be composed by integers as its elements. As we will soon see, in such cases its elements are more complex and can be represented as vectors, or as polynomials. Finite fields are often called Galois fields in honour of their discoverer. Thus, \mathbb{Z}_5 can also be called $GF(5)$ and \mathbb{Z}_2 is interchangeable with $GF(2)$. To denote a finite field whose size is a prime power, say m , we say $GF(2^m)$. Such fields, where $m > 1$, are called field extensions.

2.6 Field Extensions

We have now reached the central topic of interest in this chapter, field extensions. Field extensions are useful to give us a way to find the roots of polynomials. The kinds of problems we are interested in deal with finite fields and with polynomials defined over finite fields, although the theory can handle any kind of field. This means, for example, that if the finite field in question is $F = \{0, 1, 2, 3, 4\}$, the polynomials we can build have coefficients that can be taken *only* from this set of integers. Such polynomials are not constrained in their degree, they can be of arbitrarily high degree. The (infinite) set of all such polynomials is not a field, it is a ring with identity, denoted by $F[x]$. Since a ring needs to satisfy the closure condition under addition and multiplication, when any two polynomials are added or multiplied together the arithmetic operations on their coefficients are performed modulo (in this case) 5. This ensures closure with respect to F . If a polynomial defined in this way has no roots in the field over which it is defined, it is called **irreducible**. Polynomials whose leading coefficient (coefficient of the highest power of x) is 1 are called **monic** polynomials.

We finally come to the main theorem of interest, due to Kronecker. Let F be any field and let $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m \in F[x]$ be an irreducible polynomial over F . Then \exists a field S containing F as a subfield such that $f(x)$ has a root in S . S is the extension field, F is the base field.

Surprising as it may seem, finding such a root is *not* our main goal! We will show why the above theorem is true by relying on most of the concepts discussed so far, which is in itself quite interesting. However, in the process we will discover that the roots of $f(x)$ in the extension field have a very unexpected representation, which makes them look rather useless. Our state of confusion will be rescued by a method that shows how the algebraic structures in the extension field (the roots) are related to the algebraic structures in the polynomial ring from which $f(x)$ is taken through an elaborate mechanism [16]. This 'mechanism' sheds light on network coding techniques, highlighting how they too do not need to find any roots explicitly. In future work we will explore Gordon's technique as it makes it relatively easy to verify the abstract theory presented here with simple numerical examples. We will not have time or space in this article to delve into Galois theory, which requires another level of abstraction and a firm grasp of groups, in addition to rings and fields. The results we will discuss, however, should already

provide a sufficiently rich context to begin thinking about how to map them to security through logic, and to interaction computing through biology.

Our first step is to note that we can define an infinite number of ideals as subrings of $F[x]$. Each ideal is generated by any one of the polynomials in the ring. For simplicity let's assume that we are dealing only with irreducible polynomials over F . We could take F to be the field $GF(2) = \{0, 1\}$ without loss of generality. In other words, to prepare ourselves for the network coding examples (to be discussed in Section 3) and the application to information systems in general we can use the binary 'alphabet' as our base field.

Kronecker's theorem is proven by first establishing the following. Given a ring R and an ideal I of R , \exists a ring S and a surjective map $\theta: R \rightarrow S$ such that $\text{Ker}(\theta) = I$. This means that the polynomial of interest (and all its possible multiples taken from $F[x]$) is mapped to 0 in S .

The ideal generated by a polynomial $f(x)$ is the set of the possible products of all the elements of $F[x]$ with $f(x)$, and is denoted by $[f(x)]$. The ring S is nothing more than the factor ring

$$S = \frac{R}{I} = \frac{F[x]}{[f(x)]}, \tag{2.11}$$

which is (conceptually) built by dividing all the polynomials in $F[x]$ by the polynomial $f(x)$ and retaining only the remainders. Each different remainder represents a different coset representative for a different coset of R . How many representatives are there? How big is S ? It turns out it's not that big.

For example, if the irreducible polynomial in question is of degree 6, the remainders can, by definition, be of degree 5 at most:

$$a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5. \tag{2.12}$$

Because the coefficients are taken from $\mathbb{Z}_2 = GF(2)$ this is equivalent to a binary string of 6 bits. Thus, there is a total of $2^6 = 64$ elements in S for this example.

In other words, the required map is division of $F[x]$ by $f(x)$, modulo $f(x)$. Because this maps $F[x]$ to the ring of remainders obtained by dividing by $f(x)$, any multiple of $f(x)$ will give 0 remainder. But any multiple of $f(x)$ is the ideal $I = [f(x)]$ generated by $f(x)$. Therefore, I is mapped to the 0 element of S . Therefore, $\text{Ker}(\theta) = I$ as required.

We can now prove Kronecker's theorem. This is done in two steps: (1) we need to show that $S = R/I$ is a field; (2) we need to show that S contains a root of $f(x)$. To prove the first part we first note that each element of S is usually written as

$$I + r(x), \tag{2.13}$$

where $r(x)$ is the remainder we have been talking about. Now suppose that

$$f(x) = b_0 + b_1x + b_2x^2 + \dots + b_mx^m \tag{2.14}$$

and that $r(x) + I$ is a non-zero element of S . Because $r(x)$ is a remainder upon division by $f(x)$, necessarily the Greatest Common Divisor in $F[x]$ $GCD(r(x), f(x)) = 1$. One of the algebra basics we glossed over is the result that, in $F[x]$, $\exists s(x), t(x)$:

$$\begin{aligned} s(x)r(x) + t(x)f(x) &= GCD \\ s(x)r(x) + t(x)f(x) &= 1 \end{aligned} \quad \text{in } F[x] \tag{2.15}$$

Now the element 1 of R is mapped to the element $1 + I$ of S because clearly the number 1 can only be a remainder in a division by $f(x)$. Therefore, Eq. (2.15) in S becomes

$$s(x)r(x) + t(x)f(x) + I = 1 + I \quad \text{in } S$$

By the rules by which ideals are added and multiplied (Eq. (2.10)),

$$\begin{aligned} [s(x)r(x) + I] \oplus [t(x)f(x) + I] &= 1 + I && \text{in } S, \\ [s(x) + I] \odot [r(x) + I] \oplus [t(x) + I] \odot [f(x) + I] &= 1 + I && \text{in } S. \end{aligned}$$

But, by the definition of ideal,

$$f(x) + I = I = 0 \quad \text{in } S!$$

As a consequence,

$$[s(x) + I] \odot [r(x) + I] = 1 + I \quad \text{in } S, \tag{2.16}$$

which means that the element $[r(x) + I]$ has an inverse in S :

$$[r(x) + I]^{-1} = [s(x) + I] \quad \text{in } S. \tag{2.17}$$

Hence, S is a field.

Now we need to show that S contains a root of $f(x)$. To do this it is helpful to refer to Fig. 4, which is also taken from Allenby. In the figure we can see how the original field F is a subset of $F[x]$ since b_0 in Eq. (2.14) spans over (can take on the value of) every element of F ; of course this refers to those polynomials for which b_1 and all the higher coefficients are 0, i.e. to constant polynomials. The map θ then involves dividing all the polynomials in $F[x]$ by a particular polynomial $f(x)$, keeping only the remainders. Clearly, all the constant polynomials can only be remainders. Therefore, the field F is identically reconstructed in the factor ring S (which we have now established is a field). This is shown as $\theta(F) = F$ in the figure.

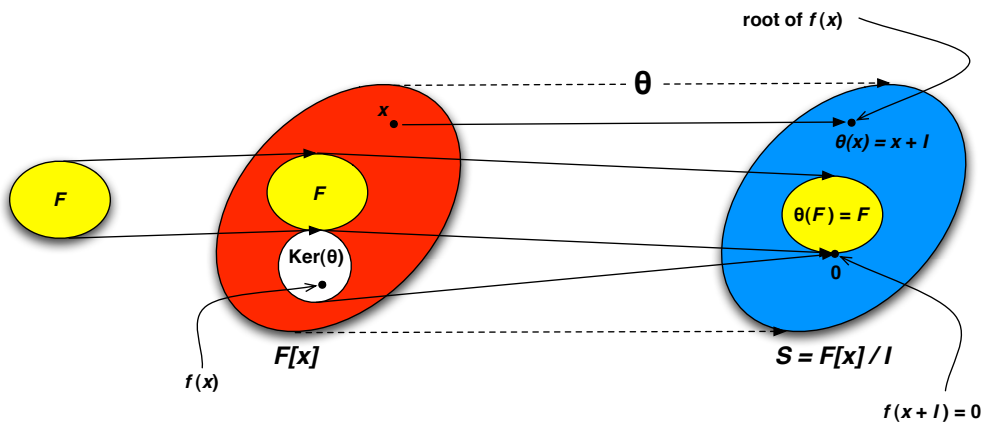


Fig. 4. Finding the root of $f(x)$

The final step in the proof is almost unremarkable and one struggles to see its significance at first. We notice that we have constructed the field S in such a way as to make $f(x)$ and its ideal map to 0. Then, necessarily, the variable x must map to the root of $f(x)$, which we will call α . How can we be sure? We know that x is of degree smaller than $f(x)$ because, if $f(x)$ were a first-order polynomial then its root would lie in F and we would have no need for all this work. So $f(x)$ is of degree 2 or higher (and it is irreducible in F). Because x is of degree smaller than $f(x)$ it is one of the remainders, and therefore it is an element of S :

$$\theta(x) = x + I = \alpha. \tag{2.18}$$

To show that it is indeed a root of $f(x)$ we substitute it into Eq. (2.13):

$$\begin{aligned}
f(\alpha) &= b_0 + b_1\alpha + b_2\alpha^2 + \dots + b_m\alpha^m \\
&= b_0 + b_1(x+I) + b_2(x+I)^2 + \dots + b_m(x+I)^m \\
&= b_0 + (b_1x+I) + (b_2x^2+I) + \dots + (b_mx^m+I) \\
&= I + (b_0 + b_1x + b_2x^2 + \dots + b_mx^m) \\
&= I + f(x) \\
&= I \\
&= 0
\end{aligned} \tag{2.19}$$

Here is where we notice for the first time that we have 'found' the root of $f(x)$ without having any idea at all about its numerical value. We understand what it is, but we do not really know it! Our understanding appears to be quite useless. To carry the point further, we have now found the extension field S but, whereas we have a very concrete idea of what F is, the nature of S escapes us. Be that as it may, we know that each element of S looks like Eq. (2.13). Our only way forward is to map $r(x)$ to S the same way we have just done for $f(x)$, that is, substituting the image of x under θ into the image of $r(x)$:

$$\begin{aligned}
I + r(x) &= I + (a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1}) \\
&= a_0 + (a_1x+I) + (a_2x^2+I) + \dots + (a_{m-1}x^{m-1}+I) \\
&= a_0 + a_1(x+I) + a_2(x+I)^2 + \dots + a_{m-1}(x+I)^{m-1} \\
&= a_0 + a_1\alpha + a_2\alpha^2 + \dots + a_{m-1}\alpha^{m-1} \\
&= r(\alpha)
\end{aligned} \tag{2.20}$$

The result tells us that each element of S is isomorphic to the remainders of $F[x]/[f(x)]$, but expressed in terms of the root we have 'found'. Interestingly, we have discovered quite a bit about the extension field of F without calculating any 'numbers'. The nature of the extension field S is in fact to be isomorphic to a vector space of dimension 2^m over the field F . In other words, since $F = GF(2)$, $S = GF(2^m)$. The different powers of the root α play the role of basis vectors. We recognise in this the same structure of the complex field \mathbb{C} , each of whose elements is of the form $a + ib$, where i is the root of the polynomial $x^2 + 1$, which is irreducible over the real field \mathbb{R} .

3 Network Coding

Our purpose in this sub-section is not to summarise results that have been known for 40 years or more in order to apply them in the manner originally intended. For that, we need only provide bibliographical references. The point of developing all this algebra, from the point of view of our research in bio-inspired computing, is not functionalist (i.e. to improve performance) in the first instance; rather, it is to understand the algebraic structure itself and then see what this structure has in common with logic on the one hand and with the DNA on the other, as will be discussed in subsequent sections. There is of course an expectation that the biological insights will ultimately bring a number of advantages, including better performance, but the first objective is to develop a common formalism for these very different domains. To this end, network coding is a very useful 'case study' because it provides a very practical context against which the abstract concepts discussed in the previous sub-sections will hopefully become easier to understand. The following discussion relies on [17], [13], and [18].

We need to introduce some basic terminology of linear block codes, upon which we will build a powerful algebraic structure by progressively introducing additional constraints. We start with the notion of a vector space. Put simply (even if perhaps too simplistically), a **vector space** is a field in more than one dimension. Each element of a vector space of dimension n is a tuple of n objects, with each object an element of a field F . An n -dimensional vector space over F is denoted by F^n . If F is the binary field \mathbb{Z}_2 , F^n has exactly 2^n elements.

Linear block codes operate by adding redundancy to a message in order to allow the original message to be reconstructed in the presence of transmission errors. Given a signal stream, we break it up into an unspecified number of message blocks of length k , taken from F^k . Thus, the message space is a vector space of dimension k and exactly 2^k elements (message blocks). To make the code unambiguous and invertible back to the original message, the mapping between the message space and the code space must be 1-1 and invertible. Therefore, the code space is a k -dimensional sub-space of F^n and also contains 2^k different elements, although each is of length $n > k$.³ The positive consequence of setting things up in this manner is that the encoding map becomes a linear transformation (i.e. a matrix multiplication) ([13], 239).⁴ Thus, we multiply each message block (row vector) by a matrix of dimensions $(k \times n)$ (k rows by n columns). The result of this multiplication for each block is a code vector of length n , called a codeword. Each codeword is sent wirelessly by the transmitter and received by the receiver, possibly with some errors. The receiver must detect these errors, correct them, and then map each codeword back to the original message block to reconstruct the original signal.

Error detection is achieved by introducing the check matrix, which is another important plot line in the discussion. Drawing on elementary linear algebra, Theorem 3.4 in [17] states that, "For any $k \times n$ matrix G over $GF(q)$ with k linearly independent rows, there exists an $(n - k) \times n$ matrix H over $GF(q)$ with $(n - k)$ linearly independent rows such that for any row g_i in G and any h_j in H , $g_i \cdot h_j = 0$. The row space of G is the null (dual) space of H , and vice versa." Since the rows of G provide a basis for the code space, the same rule applies to any linear combination of its rows, i.e. to any codeword. Therefore, any error can be easily detected as a non-zero result when multiplying a received codeword by H^T . Appropriate error correction methods then can be applied, as discussed in the literature. Figure 5 provides a schematic summary of these concepts.

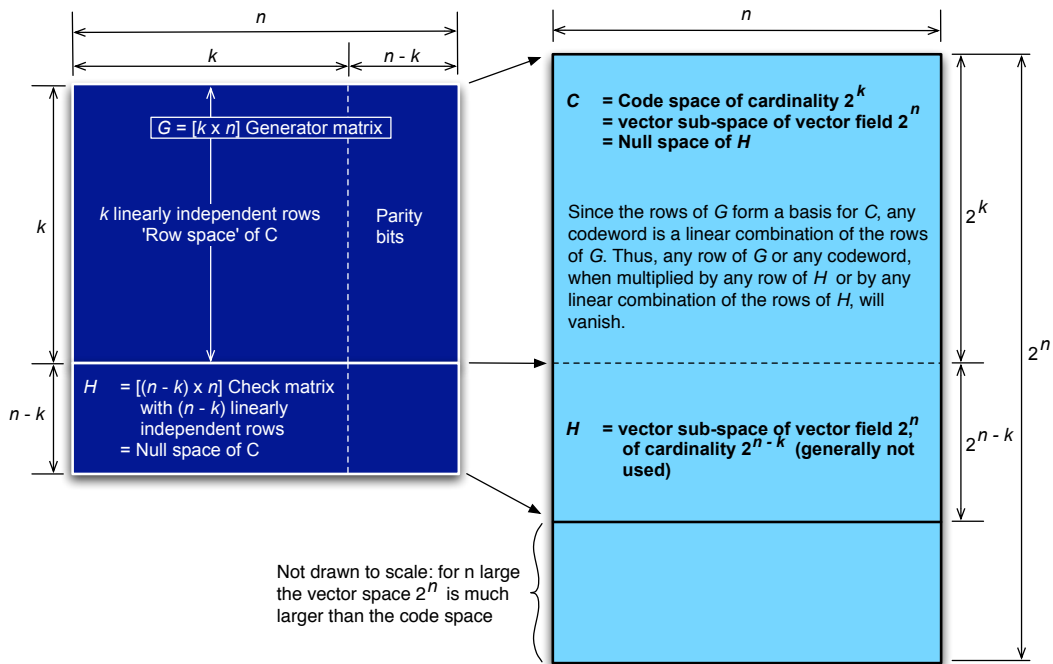


Fig. 5. Schematic summary of linear block code

We now start connecting to abstract algebra by noticing that in network coding applications the focus of attention is not a polynomial whose roots we need to find through the generation of a field extension as a factor ring (Galois's original problem) but, rather, the factor ring itself. Furthermore, the

³ This is analogous to a slanted 2D plane immersed in 3D space: each point on the plane is defined by 3 coordinates, but the plane itself remains a 2D subspace.

⁴ As another example of the wonderful generality of algebraic structures, note that a linear transformation is also a homomorphism of vector spaces ([19], 5).

generation of a field extension is a necessary step in finding the roots of an *irreducible* polynomial. As it happens a polynomial does not need to be irreducible to form an ideal, and therefore it does not need to be irreducible to form a factor ring. An additional point that should help connect the work in the previous sub-section to this discussion is that the concept of ideal can be applied, in a sense, recursively. In other words, a factor ring is obtained by ‘dividing’ a ring $GF(2)[x]$ by an ideal, say $[(x^n - 1)]$. The resulting factor ring can also contain ideals. It is these latter ideals that we are mainly going to use. In the discussion of cyclic codes that follows we identify the vector space F^n with the factor ring, and the code space with one of its ideals (see Figure 6).

A **cyclic code** is a linear block code with the characteristic that every cyclic shift of any of its codewords yields another codeword. A **cyclic shift** is the n -tuple obtained by shifting every component to the right (or left) and wrapping the last component back to the front of the list. When working with cyclic codes, it is useful to associate each codeword of length n with a polynomial of degree $n - 1$ or less, whose coefficients are the coordinates of the code vector:

$$c(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1}. \tag{5.21}$$

$c(x) \in F[x]$ is called the code polynomial. The set of 2^k code polynomials of a code C is a subset of the factor ring $GF(2)[x]/[(x^n - 1)]$, which is not a field since $(x^n - 1)$ is factorisable. Incidentally, from this fact we deduce that whereas every finite field contains a number of elements that is equal to an integer power of a prime number, the converse is not true: a set that contains a number of elements that is equal to an integer power of a prime number is not necessarily a field. To clarify, the factor ring $GF(2)[x]/[(x^n - 1)]$ is not a field because not every one of its elements has an inverse (due to the fact that the GCD of each element and $(x^n - 1)$ is not necessarily 1). From this point of view the factor ring is a one-dimensional object. But the same object can also be described as a ‘vector field’ of dimension n , F^n . From this second point of view the use of the term ‘field’ refers to the fact that the *ground* field $\{0, 1\}$ is indeed a field. We notice that in this factor ring, by construction, operations between its elements are performed modulo $(x^n - 1)$. The reason for creating the factor ring using $(x^n - 1)$, rather than an irreducible polynomial of the same degree, is that in the latter case the factor ring would be a field, and therefore its only ideals would be 0 and itself. Why this matters is shown next.

In the polynomial ring $GF(2)[x]/[(x^n - 1)]$ the cyclic shift of a codeword by k is equivalent to multiplication modulo $(x^n - 1)$ of the corresponding code polynomial by x^k . This fact in itself seems unremarkable until we realise that an arbitrary polynomial $a(x)$ can be considered as a linear combination of cyclic shifts which, according to the definition, yields another $c(x)$ that still belongs to the same code C . Because we already knew that C is a sub-space of F^n (or a subring of the factor ring), we know that it is closed under addition: adding two vectors yields a third vector in the same plane the two vectors define, even if this plane is immersed in a 3D space. As a consequence, the code C satisfies the ideal test and is in fact an ideal of the ring $GF(2)[x]/[(x^n - 1)]$. Because it is a subset, the Hamming distance between codewords is generally greater than zero and can be ensured to be greater than zero, as will be shown below, which is what makes error correction possible.

The ring of polynomials $GF(2)[x]$ is a Principal Ideal Domain, which means that every one of its ideals is principal. A **principal ideal** is an ideal that is entirely generated by a single one of its elements. For example, the element 2 of \mathbb{Z} generates the ideal $[2]$ of \mathbb{Z} (the even numbers). A useful theorem ([13], 245) states that if any ideal I in a ring R can be generated by a single element, then the same is true of any ideal of any of the factor rings R/I . This is enough for our purposes, but notice that it is a little weaker than to say that if R is a PID, then R/I is also a PID. This latter statement is true only if polynomial that generates I is irreducible, in which case as we saw the factor ring R/I is actually a field. If R is a PID but the polynomial that generates I is reducible (our case), then the factor ring R/I is not an integral domain; therefore, it cannot be a PID even though every one of its ideals is in fact principal. All of this matches exactly what happens with the ring of integers modulo an integer n .

In our case we are working with $(x^n - 1)$. The factor ring $GF(2)[x]/[(x^n - 1)]$ is not a field and is not a PID either. However, every one of its ideals *is* principal ([13], 245). This implies that our code C is entirely generated by a single polynomial $g(x)$, aptly called the generator polynomial. The **generator polynomial** is the monic polynomial of least degree (for a particular code) that belongs to C and from

which every element of C can be generated through multiplication by elements of $GF(2)[x]$ modulo $(x^n - 1)$. There is a unique such polynomial for any ideal of $GF(2)[x]/[(x^n - 1)]$, and each such instance divides $(x^n - 1)$ ([18], 32). It follows that to generate all the possible cyclic codes for a given value of n we need to find all the irreducible factors of $(x^n - 1)$. The possible generator polynomials $g(x)$ are all the possible divisors of $(x^n - 1)$, formed as single irreducible factors or as products of these irreducible factors. If $g(x)$ is chosen of degree $n - k$, a linear code results and the generator matrix can be formed starting from the simple statement $m(x)g(x) = c(x)$, where $m(x)$ is a polynomial of degree $k - 1$ representing a message block. Writing out $m(x)$ we see that each term will cause a cyclic shift in $g(x)$:

$$\begin{aligned} c(x) &= m(x)g(x) \\ &= (m_0 + m_1x + \dots + m_{k-1}x^{k-1})g(x) \\ &= m_0g(x) + m_1xg(x) + \dots + m_{k-1}x^{k-1}g(x) \end{aligned} \quad (5.22)$$

Thus, for $g(x) = g_0 + g_1x + g_2x^2 + \dots + g_{n-k}x^{n-k}$, dropping the x , the generator matrix looks as follows:

$$\begin{aligned} c &= [m_0m_1\dots m_{k-1}] \cdot \\ &\begin{bmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & 0 & g_0 & g_1 & \dots & g_{n-k} & \dots & 0 \\ \dots & & & & & & & \\ 0 & 0 & 0 & \dots & g_0 & g_1 & \dots & g_{n-k} \end{bmatrix} \end{aligned} \quad (5.23)$$

Furthermore, there exists a polynomial $h(x)$ such that $g(x)h(x) = (x^n - 1)$, which gives rise to the check matrix.

Figure 6 summarises the concepts discussed so far. The width of the rectangles represents the dimension of the vector space or the largest possible number of terms of the polynomial representation of the code. $g(x)$ is shown in red to emphasise that it is one of the code polynomials, the one of least degree. The number of elements in the factor ring is 2^n . A cyclic code is an ideal of this ring, and therefore a subset. Before reaching the next kind of codes (BCH codes) we need to discuss a particular aspect of the check matrix.

The Roots of $g(x)$ The discussion now enters a territory where abstract algebra becomes increasingly entangled with linear algebra. Although the immediate benefit is just a better understanding of network coding, it seems worthwhile to chart this territory as carefully as possible because the algebraic machinery of network coding that we are slowly unravelling provides a relatively concrete example of a mathematical modelling framework that is likely to be relevant to the DNA and to interacting state machines. The goal of the discussion, therefore, remains to *understand*, rather than to utilise network coding results that have been known for decades and that are not, in and by themselves, likely to be directly applicable to biologically-inspired computing.

Purser ([18], 35) points out that since $g(x)$ is the generator polynomial of a cyclic code, it necessarily divides all the codewords, in addition to dividing $(x^n - 1)$. Therefore, the roots of $g(x)$ are also (some of) the roots of any codeword polynomial $c(x)$. Let α be one such root. Then, for any

$$c(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1}, \quad (5.21)$$

$$c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + \dots + c_{n-1}\alpha^{n-1} = 0 \quad (5.24)$$

As a consequence, the row vector

$$(1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{n-1}) \quad (5.25)$$

must belong to the null space of G , i.e. it must be a row of H . This argument is a little ‘sneaky’ because in Eq. (5.21) we are using the code polynomial as just one convenient way to represent a code vector, where the powers of x play the role of basis vectors, whereas in Eq. (5.24) we have switched completely

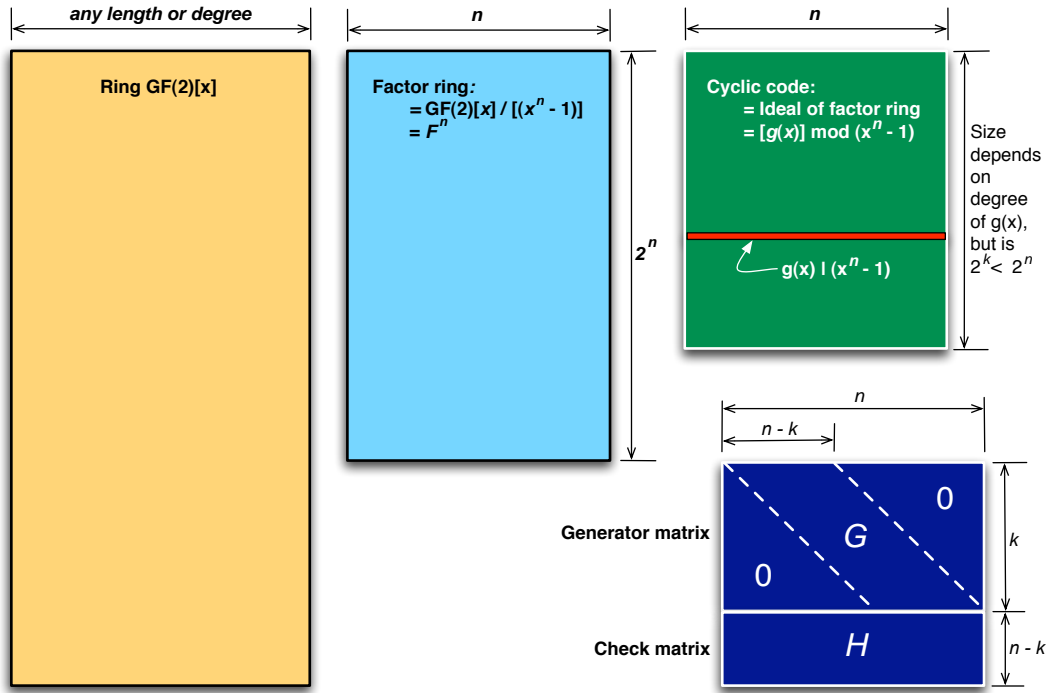


Fig. 6. Schematic summary of cyclic code

to the algebraic perspective and are treating the polynomial as a function of x . None-the-less we have done nothing wrong and the conclusion (5.25) stands.

Notice also that, whereas when describing linear block codes and cyclic codes we worked mainly with factor rings and their ideals, we appear to have brought the roots of polynomials back into the discussion. To avoid possible confusion let's clarify what is happening. Focus for the moment on an irreducible $g(x)$. Since we know that it is of degree $m = n - k$, the roots that we are talking about must lie in a field extension $GF(2^m)$. We must emphasise that this field is different and additional to any other fields or rings we have been talking about until now. If $g(x)$ is irreducible its roots in $GF(q^m)$ are expressible as powers of the first one, as follows:

$$\alpha_i = \alpha_1^{q^{i-1}} \tag{5.26}$$

For $GF(2)$ this results in the sequence

$$\alpha, \alpha^2, \alpha^4, \alpha^8, \dots, \alpha^{2^{m-1}} \tag{5.27}$$

Therefore, a check matrix of $n - k$ rows by n columns can be constructed as follows:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & (\alpha^2) & (\alpha^2)^2 & \dots & (\alpha^2)^{n-1} \\ 1 & (\alpha^4) & (\alpha^4)^2 & \dots & (\alpha^4)^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & (\alpha^{2^{m-1}}) & (\alpha^{2^{m-1}})^2 & \dots & (\alpha^{2^{m-1}})^{n-1} \end{bmatrix} \tag{5.28}$$

The check matrix also gives a way to determine the minimum distance of a code. In a linear code the minimum distance between any two codewords equals the minimum weight (the weight is the number of 1s in a codeword) among all the codewords. Furthermore, as indicated in Figure 5, multiplying any codeword (or any row of G) by H^T equals zero. As a consequence, codewords of weight w correspond to dependence relations among sets of w columns of H (or rows of H^T) ([13], 243). Therefore, the minimum weight among all codewords, which equals the smallest distance between any two codewords of a code, is equal to the minimum number of linearly dependent columns of H .

In building cyclic codes it is easy to set their length and dimension. However, the spacing (in terms of Hamming distance) of the $c(x)$ codewords within the vector space F^n is not necessarily uniform, which means that the minimum distance is both uncertain and not easy to find, especially as n becomes large. BCH codes are also cyclic codes, but they introduce an additional constraint that makes it possible to specify the minimum distance d at the outset. The fact that, as already mentioned, the vector field F^n has the same cardinality as $GF(2^n)$ may at first seem to imply that F^n is a field, which would be potentially confusing since we just spent considerable effort to prove that F^n (in the present context) is a ring and not a field. Isomorphism, however, requires more than cardinality. It also requires that the elements map 1-1 and that the same map satisfy Eq. (5.6). It is worthwhile explaining this point carefully because it will make it easier to understand the BCH codes, so let us introduce an example.

Assume that, using the notation of Eq. (5.5), $R = GF(2)[x]/[(x^4 - 1)]$ and $S = GF(2)[x]/[(x^4 + x + 1)]$. Since we are working with the ground field $GF(2)$ for the coefficients it does not really matter whether we use $+$ or $-$. The first of these polynomials is reducible, the second irreducible. They both give rise to a factor ring of the same cardinality as $GF(2^4)$, composed of all possible polynomials of degree 3 or less: there are 16 of them, including the ground field elements 0 and 1. More interestingly, these are necessarily *exactly the same* polynomials. Thus, we could envisage a 1-1 map θ between the elements of these two sets—in fact, the identity map. The point is that this map is not a homomorphism, and therefore not an isomorphism either, as we now show.

R is not a field while S is because these two sets are the remainders of different ideals. Therefore, if we take any two elements of R and multiply them together, for example, the result may be of degree higher than 3. Thus, it would need to be divided by $(x^4 - 1)$, keeping the remainder, which is another element of R . If the *same* two elements are taken from S , now their product will need to be divided by $(x^4 + x + 1)$, keeping the remainder. This second remainder will be an element of S that is *different* to the corresponding remainder we got in R . Therefore Eq. (5.6b) is not satisfied and we don't have a homomorphism. As a consequence the map is not an isomorphism even if it could be the identity map as in this example. In general, therefore, because $GF(2^n)$ is a vector field of *remainders*, it cannot be considered independently of the ideal that generated it as a factor ring. When $GF(2^n)$ is the splitting field for $(x^n - 1)$, i.e. the field that contains all of the roots of all of the irreducible factors of $(x^n - 1)$, it will contain several ideals, each of which is generated by a single element (which could be the product of these irreducible factors) and each of which serves as a possible cyclic code.

But let us go back to the generator polynomial $g(x)$ and to its roots in the extension field $GF(2^m)$. Figure 7 shows how the discussion of $g(x)$ and its roots concerns quite a different field, even if, in fact, these same roots will also solve any of the codeword polynomials. We know that if α is a primitive element of $GF(2^m)$ all the elements of this field extension can be represented as successive powers of this root. This represents none other than the multiplicative group of $GF(2^m)$, i.e. $GF(2^m) \setminus \{0\}$. We also know that each of the elements of $GF(2^m)$ can also be represented as an m -dimensional vector over $GF(2)$, or as a polynomial in α over the same ground field and of degree at most $m - 1$. It is possible to continue the discussion in general terms, but the formalism becomes rather overbearing. It is more effective to drop down to a specific example in order to communicate ideas whose general applicability is in any case still visible and easily understood.

For example, take

$$g(x) = x^4 + x + 1, \quad (5.29)$$

as the generator polynomial for a code of length 15. Thus, $m = 4$, $n = 15$, and $k = 11$. To say that α is a root of $g(x)$ is equivalent to saying that

$$\alpha^4 = \alpha + 1. \quad (5.30)$$

Applying this fact to generate the 14 roots of this field extension in polynomial form from the same roots expressed as successive powers of α is equivalent to constructing the factor ring modulo $g(x)$ (this is

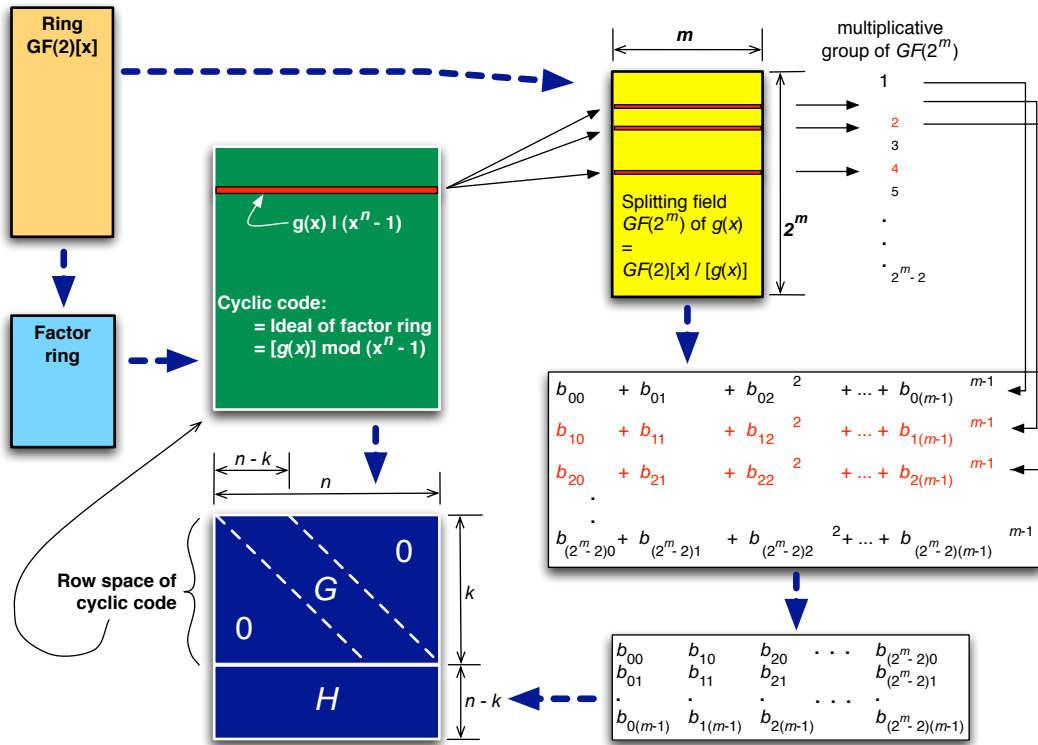


Fig. 7. The roots of $g(x)$

shown in general form on the right-hand side of Figure 7):

$$\begin{aligned}
 (\alpha^0 &= 1) \\
 \alpha^1 &= \alpha \\
 \alpha^2 &= \alpha^2 \\
 \alpha^3 &= \alpha^3 \\
 \alpha^4 &= 1 + \alpha \\
 \alpha^5 &= \alpha + \alpha^2 \\
 \alpha^6 &= \alpha^2 + \alpha^3 \\
 \alpha^7 &= 1 + \alpha + \alpha^3 \\
 \alpha^8 &= 1 + \alpha^2 \\
 \alpha^9 &= \alpha + \alpha^3 \\
 \alpha^{10} &= 1 + \alpha + \alpha^2 \\
 \alpha^{11} &= \alpha + \alpha^2 + \alpha^3 \\
 \alpha^{12} &= 1 + \alpha + \alpha^2 + \alpha^3 \\
 \alpha^{13} &= 1 + \alpha^2 + \alpha^3 \\
 \alpha^{14} &= 1 + \alpha^3 \\
 (\alpha^{15} &= 1)
 \end{aligned}
 \tag{5.31}$$

Since in this example $m = 4$, these roots can now be substituted into the 4 rows of H in Eq. (5.28), after reducing the powers modulo 15. This is done by treating each root as a 4×1 column vector. As a consequence H becomes a 16×15 matrix. As discussed by Purser ([18], 36), since for an irreducible $g(x)$ (actually primitive in this example) the higher roots are given as powers of the first (Eq. (5.27)), the second, third and fourth rows of H in Eq. (5.28) do not add any information. For this example, direct substitution does in fact show that of the 16 rows of H thus constructed only 4 are linearly independent. We might as well, therefore, only retain the first row of H in Eq. (5.28) which, since n was chosen to be 15 in this example, corresponds to all the vectors shown in Eq. (5.31). These can in fact be directly

entered as tuples over $GF(2)$ as follows:

$$H = \begin{bmatrix} 1000100110101111 \\ 010011010111100 \\ 001001101011110 \\ 000100110101111 \end{bmatrix} \begin{matrix} \alpha^0 \\ \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{matrix} \quad (5.32)$$

where the powers of α to the right of each row of the matrix are shown just to help recognise the patterns of 1s and 0s corresponding to Eq. (5.31). This matrix is shown in general form at the bottom-right of Figure 7. The choice of m such that $n = 2^m - 1$ makes this example a Hamming code, for which the

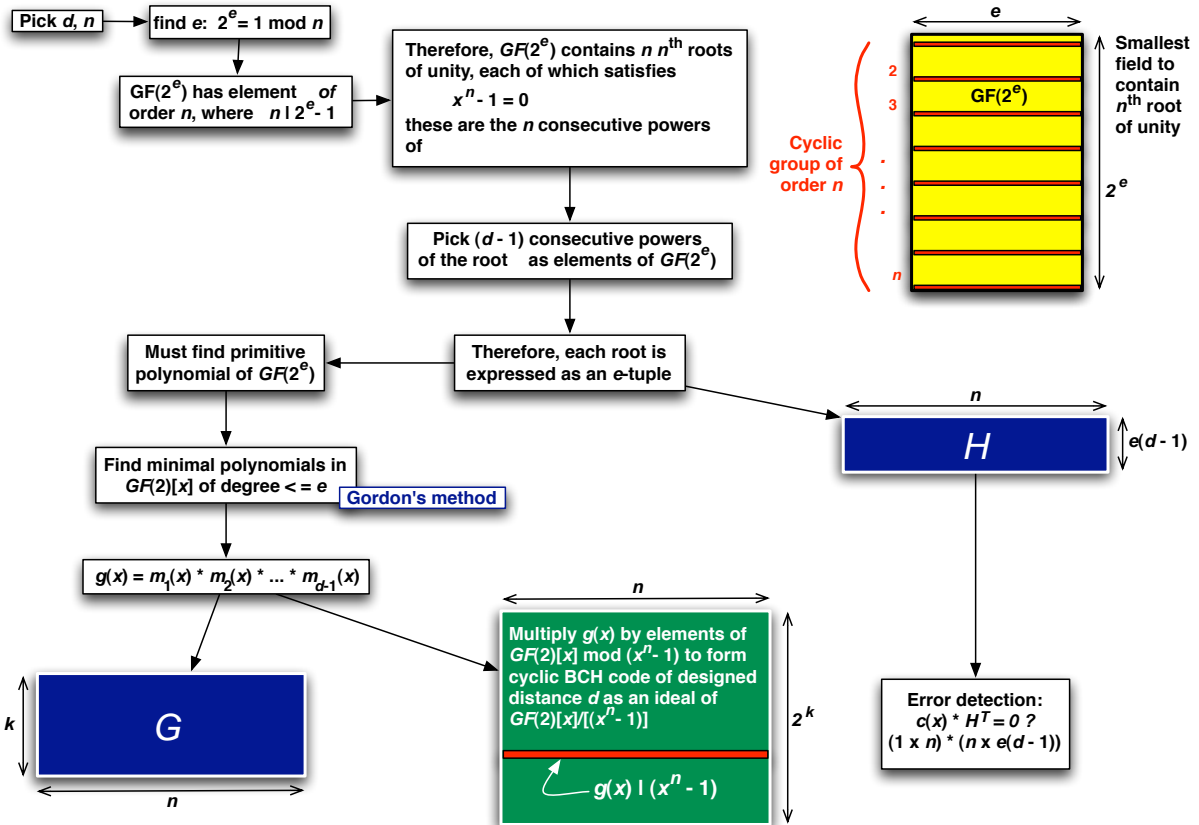


Fig. 8. Schematic summary of BCH code

minimum distance is $d = 3$ (and therefore this code can only correct 1-bit errors per codeword, at most). To see this we just need to note that all the elements of $GF(2^m)$ (except for 0) fit in H as distinct columns. The sum of any two, however, necessarily gives another column from the same set due to the closure of $GF(2^m)$ with respect to addition. Therefore, the largest number of linearly independent columns is 2 and the smallest number of linearly dependent ones is 3. As proven a few paragraphs above, the minimum distance is therefore 3. If we wanted a greater distance for the same length n , still using an irreducible $g(x)$ (necessarily of higher degree), the code rate k/n would suffer. It turns out that if, on the other hand, $g(x)$ is not irreducible but is composed of 2 or more irreducible factors, the rows of H will be formed by more than one root. As a consequence, the greatest number of linearly independent columns of H is more likely to be larger than 2, leading to a greater minimum distance. The code rate is not necessarily going to be the same, which highlights how optimisation and trade-offs are required when mathematical results encounter engineering applications—which however is not our main concern here.

As explained by Purser ([18], 47), the above observation motivated the development of the BCH codes, for which the generator polynomial is the least common multiple of the minimal polynomials of $d - 1$ consecutive powers of a primitive n^{th} root of unity, for a desired minimum distance d . A **primitive**

n^{th} root of unity in a field F is an element a whose order in the multiplicative group of F is precisely n . As a consequence, a and all of its powers are roots of the equation $x^n - 1 = 0$. By the definition of order of a group element, all the powers of a up to a^{n-1} are distinct. Therefore, all the n roots of $x^n - 1 = 0$ are generated by a and its powers. From these we can choose $d - 1$ consecutive ones.

Figure 8 shows the steps of the process required to develop a BCH code. For a given code length n we need to find a primitive n^{th} root of unity, and the smallest field that contains $GF(q)$ and a primitive n^{th} root of unity is $GF(q^e)$, where e is the order of q mod n ([13], 248). If n and q are two coprime integers, the **order** of q mod n is the smallest positive integer e for which $q^e = 1 \pmod{n}$; i.e. e is the order of q in the multiplicative group \mathbb{Z}_n . For example, if $n = 10$ and $q = 3$, $e = 4$. In other words, the smallest field that contains a primitive 10^{th} root of unity over the base field $GF(3)$ is $GF(3^4)$. As another example, for $GF(2)$ and $n = 21$ $e = 6$, because $2^6 = 64 = 1 \pmod{21}$. In practice, one tends to use a value of $n = 2^e - 1$: $n = 127$ for $e = 7$, or $n = 255$ for $e = 8$. These codes are called, unsurprisingly, primitive BCH codes since the n^{th} root of unity β in such a case is a primitive element of $GF(2^e)$.

Figure 8 shows how the construction of H is similar to what we just discussed above. To find $g(x)$ one first needs to find a primitive polynomial of $GF(2^e)$, and then several minimal polynomials for the $d - 1$ roots. Finding a primitive polynomial is a task that needs to be done only once for each value of e . In addition, any one among the set of primitive polynomials for a particular value of e will do, since the others give rise to isomorphic fields. Finding the minimal polynomials, on the other hand, can be laborious. Here is where Gordon's method becomes useful. We will mention it briefly, mainly to highlight another interesting structural aspect of the algebra of network coding.

An Observation on Gordon's Method Gordon's method [16] is relevant to BCH codes and is concerned with finding the minimal polynomials given the knowledge of their roots in the extension field $GF(2^e)$. These roots can be expressed as consecutive powers of a primitive element or as the remainders of $f(x)$ in the factor ring $GF(2)[x]/[f(x)]$, where $f(x)$ is the irreducible primitive polynomial that generates the extension field $GF(2^e)$. Because the degree of these minimal polynomials can be at most e , there are several in any given extension field with 2^e roots. Gordon realised that, for the same reason, each minimal polynomial can also be expressed modulo the same primitive polynomial. In other words, each minimal polynomial can be expressed as one of the elements from the set of its own roots. This makes the determination of the minimal polynomials trivially simple and ideally suited for low-power space probes, where every bit of memory and CPU cycle counts. Even more surprisingly, however, we cannot avoid the conclusion that a minimal polynomial and one of its roots may actually have the same polynomial form! It is not clear whether this fact has any significance, but such examples of structural invariance cannot help but stimulate our curiosity. This particular example is reminiscent of conformal invariance of differential equations under the action of a Lie group of transformations or, to a smaller extent, of the concepts of eigenvector in oscillatory systems or the renormalisation group of statistical physics.

In this section we have used network coding to give a flavour for the richness of algebraic structure, which may at first seem quite far removed from biology. As it happens, the work reported in [5,20] indicates otherwise and motivates further investigation in Lie algebras.

We now shift the perspective from algebra to logic. There are a few angles from which one can look at the connections between algebra and logic, for instance comparing propositional logic and Boolean algebras; or first-order logic and cylindric algebras. We are also investigating the model checking point of view applied to automata and to the underlying semigroup properties. Finally, temporal logics, timed automata and their potential connections to dynamic(al) systems through Lie algebras.

4 Logic, Algebra, and Security

Having presented a general overview of abstract algebra and its very basic concepts, this section is going to establish a link between the disciplines of logic, algebra, and security.

For this purpose we start by defining propositional and first-order logic. Section 2 should help see the strong similarities between logic and algebra, especially when comparing Galois fields and propositional

logic. We will attempt to show how the different logics find their counterparts in the realm of abstract algebra. With the informal extension of the basic concepts of logic and algebra to temporal logic and quantifier algebras, respectively, we will bridge from algebra to security. This not entirely obvious bridge is emphasised in the final sub-section, where we show that the intentional application of temporal logic and its wide use in the realm of computer science is process and program analysis and is therefore directly linked to security.

4.1 Propositional and First-Order Logic

In this section we want to give a short introduction to propositional and predicate logic, two important fields of the research area of symbolic logic. Of course, this introduction cannot be exhaustive and we will refer the reader to appropriate literature.

The following sub-section will be the basis for Sub-sections 4.2 and 4.3. However, for the experienced reader who is familiar with the basic concepts of these logics we recommend to proceed directly to sub-section 4.2.

Propositional Logic Propositional logic is often called sentential logic. Both names account for the nature of this logic since it uses sentences, thus *sentential*, which can be thought of as *propositions*. The classical propositional logic studies the effects of propositional connectives, such as *and* and *or*, which can be used to form new sentences out of atomic sentences. To formalise this quantitative description we define the syntax of propositional logic before defining its semantics.

For the following definitions we assume that \mathcal{P} is an infinite (countable) alphabet of propositional letters. Additionally we denote \top and \perp for the values *true* and *false* as it is usually done in classical two-valued logic. Based on these assumptions we define propositional atomic formula and propositional formula.

Definition 1. *An atomic formula is a propositional letter, \top or \perp .*

Please note that \top and \perp are currently only symbols without any interpretation.

Definition 2. *The set of propositional formula is the smallest set \mathcal{P} such that*

1. *If A is an atomic formula, $A \in \mathcal{P}$*
2. *$X \in \mathcal{P} \Rightarrow \neg X \in \mathcal{P}$*
3. *If \circ is a binary symbol, then $X, Y \in \mathcal{P} \Rightarrow (X \circ Y) \in \mathcal{P}$*

These definitions are sufficient to define the syntax of propositional logic. Due to space constraints, we skip numerous theorems and other definitions that facilitate the handling of propositional logic. Instead, we now define its semantics. To do so we first have to define so-called truth values, which can be mapped to our previously defined truth symbols. For this purpose we define the set $\mathcal{T} = \{0, 1\}$. Looking at Definition 2 we additionally have to define the operation on set \mathcal{T} represented by the operation symbols \neg and \circ .

There are some similarities and differences between these definitions and the starting point of abstract algebra. \mathcal{P} is analogous to an infinite set of variables whose values are taken from the field $\mathbb{Z}_2 = \mathcal{T}$. In abstract algebra we define more complicated 'objects' over a particular field. These objects are called polynomials. So we can have polynomials over \mathbb{Z}_2 or over $\mathbb{Z}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$. In propositional logic we do not actually have objects as complicated as polynomials, we have sets of propositions that are connected by various binary or unary operators, forming the 'sentences' mentioned above.

For the operational symbol \neg we define the mapping $\neg: \mathcal{T} \rightarrow \mathcal{T}$ by $\neg(t) = f$ and $\neg(f) = t$. Defining the binary connectives is more complicated as there are 16 possible mappings. We can enumerate the definitions of all connectives in Table 2.

After defining the basic operations of propositional logic we can now define the so-called *Boolean valuation*, which is an appropriate mapping for propositional formula.

		Connectives															
		\perp	\wedge	\supseteq	L	\subseteq	R	\neq	\vee	\downarrow	\equiv	$\neg R$	\subset	$\neg L$	\supset	\uparrow	\top
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 2. Definition of propositional logic connectives

Definition 3. A Boolean valuation is a mapping v from the set of propositional formula to the set \mathcal{T} meeting the conditions

1. $v(\top) = 1; v(\perp) = 0$
2. $v(\neg X) = \neg v(X)$
3. $v(X \circ Y) = v(X) \circ v(Y)$, for any binary operations \circ

The valuation of logic sentences and the evaluation of polynomials are both important and help us relate the theoretical concepts to observable or experiential reality. However, as will be discussed below in the sub-section on algebraic logic, our concern is less with the kind of value the elements of our set can assume, and more with the structural relationships within and between sets. This is the essence of algebra in its most abstract form, whose role as a unifying framework we are attempting to extend also to biology.

For example, Table 2 can be compared to Table 1. Although at first these seem very different, it turns out that all the binary connectives shown in Table 2 can be expressed in terms of only two connectives, \wedge (multiplication) and \neq (addition mod 2). The additional structure shown by Table 2 is analogous in logic to a field extension $GF(2^4)$ over the base field $GF(2)$.

For compliance with existing literature on propositional logic we additionally define *interpretation* and *model*.

Definition 4.

- A Boolean valuation is also called an interpretation.
- An interpretation v of a propositional formula F with $v(F) = 1$ is also called a model.
- The model of a set \mathcal{F} of propositional formula is an interpretation v with $v(F) = 1, \forall F \in \mathcal{F}$.
- \mathcal{F} is called a theory if all possible interpretations are models of \mathcal{F} .

Please note that the \neg and \circ in this definition denote symbols and operations. The appropriate meaning is defined by the context in which they are used. Additionally we note that the Boolean valuation is defined over all propositional logic connectives listed in Table 2.

With Boolean valuation in hand we can now take an arbitrary formula of propositional logic and define a valuation for it. As an example, we consider the formula $\neg(\neg(P \subset Q) \wedge \neg R)$. We now want to find the Boolean valuation v of this formula and assume that $v(P) = 1, v(Q) = 1, v(R) = 0$. One can show that such a valuation exists and that it is unique by definition. Here, we only want to show which valuation steps to take.

$$\begin{aligned}
 v(\neg(\neg(P \subset Q) \wedge \neg R)) &= \neg v(\neg(P \subset Q) \wedge \neg R) \\
 &= \neg(\neg v((P \subset Q) \wedge \neg R)) \\
 &= \neg(\neg(v(P \subset Q) \wedge v(\neg R))) \\
 &= \neg(\neg((v(P) \subset v(Q)) \wedge \neg v(R))) \\
 &= \neg(\neg((1 \subset 1) \wedge \neg 0)) \\
 &= \neg(\neg(1 \wedge 1)) \\
 &= \neg(\neg 1) \\
 &= \neg 0 \\
 &= 1.
 \end{aligned}$$

For now, these definitions should be sufficient to understand the general concept of propositional logic.

First-Order Logic As we have seen above, propositional logic can be used to build formula which represent propositions. They can be evaluated using Boolean valuation. However, propositional logic is unable to derive valid arguments which respect the internal structure of a proposition.

For this reason first-order logic (FOL) replaces the pure propositional letters by predicates which can have arguments. Thus, first order logic is often called predicate logic. Due to the introduction of variables FOL also supports quantifiers to bind variables.

As in the last subsection we will shortly introduce the syntax of first-order logic, also called first-order language. After introducing the general concept of a *model* we define the semantics of FOL.

Definition 5. A first-order language $L(\mathcal{R}, \mathcal{F}, C)$ is determined by specifying

1. A finite or countable set \mathcal{R} of relation symbols (predicate symbols) each of which has associated a positive integer n with it. n indicates the arity⁵ of the predicate.
2. A finite or countable set \mathcal{F} of function symbols each of which has associated a positive integer m with it. m indicates the arity of the function.
3. A finite or countable set C of constant symbols.

Since we have finer granularity than in propositional logic, we have to define so-called *terms* before we can take the next step and define first-order *formula*, which are analogous to propositional logic formula. For this purpose we have to introduce *variables*. They represent elements of a set which we will later call domain.

Definition 6. The family of terms of $L(\mathcal{R}, \mathcal{F}, C)$ is the smallest set meeting the following conditions:

1. Any variable is a term of $L(\mathcal{R}, \mathcal{F}, C)$.
2. Any $c \in C$ is a term of $L(\mathcal{R}, \mathcal{F}, C)$.
3. If $f \in \mathcal{F}$ has arity n and t_1, t_2, \dots, t_n are terms of $L(\mathcal{R}, \mathcal{F}, C)$, then $f(t_1, t_2, \dots, t_n)$ is a term of $L(\mathcal{R}, \mathcal{F}, C)$.
4. We denote $\mathcal{T}_{L(\mathcal{R}, \mathcal{F}, C)}$ as the set of terms of $L(\mathcal{R}, \mathcal{F}, C)$.

Terms are *closed* if they contain no variables.

Based on the terms defined above we can now define formula. You will recognise that the definition is very similar to the definition of formula in propositional logic. The main difference is the use of terms instead of propositions and the introduction of the universal (\forall) and existential (\exists) quantifiers. We again define atomic formula first.

Definition 7. An atomic formula of $L(\mathcal{R}, \mathcal{F}, C)$ is any string of the form $R(t_1, t_2, \dots, t_n)$ with $R \in \mathcal{R}$ and $t_1, t_2, \dots, t_n \in \mathcal{T}_{L(\mathcal{R}, \mathcal{F}, C)}$

Based on atomic formula we define the family of formula of $L(\mathcal{R}, \mathcal{F}, C)$.

Definition 8. The family of formula of $L(\mathcal{R}, \mathcal{F}, C)$ is the smallest set meeting the following conditions

1. Any atomic formula of $L(\mathcal{R}, \mathcal{F}, C)$ is a formula of $L(\mathcal{R}, \mathcal{F}, C)$
2. If A is formula of $L(\mathcal{R}, \mathcal{F}, C)$ so is $\neg A$
3. For a binary connective \circ , if A and B are formula of $L(\mathcal{R}, \mathcal{F}, C)$ so is $(A \circ B)$.
4. If A is a formula of $L(\mathcal{R}, \mathcal{F}, C)$ and x is a variable, then $(\forall x)A$ and $(\exists x)A$ are formula of $L(\mathcal{R}, \mathcal{F}, C)$.

⁵ In logic, mathematics, and computer science, the arity of a function or operation is the number of arguments or operands that the function takes. The arity of a relation is the number of domains in the corresponding Cartesian product. In this research we focus on the Cartesian product of a set with itself ($A \times A$). For example, the arity of the addition operation is 2, which means that addition is a binary operation, or that it takes 2 arguments. (<http://en.wikipedia.org/wiki/Arity>)

By using the syntax defined above and defining \mathcal{R} , \mathcal{F} , and \mathcal{C} appropriately we can specify the same formula that we find in popular theories such as set theory. Theories using this first-order language are also called first-order theories.

But with the syntax alone, the language is rather useless as it has no meaning. Thus the next step is to define a semantics. However, compared to propositional logic the definition of first-order logic semantics is very complicated as we are facing variables, functions, relations, and quantifiers. To slightly simplify the definition we will first introduce the concepts of *models*, which implicitly define *domains* and *interpretations*.

Definition 9. A model for the first-order language $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ is a pair $\mathcal{M} = \langle \mathcal{D}, I \rangle$ with

\mathcal{D} is non-empty set, called a domain of \mathcal{M}

I is a mapping, called an interpretation that associates

1. To every $c \in \mathcal{C}$, some member $c^I \in \mathcal{D}$
2. To every $f \in \mathcal{F}$, some function $f^I : \mathcal{D}^n \rightarrow \mathcal{D}$
3. To every $P \in \mathcal{R}$, some relation $P^I \subseteq \mathcal{D}^n$

As we can see, models do not cover variables. But as terms in first-order language are based on variables we also need to *assign* elements from a domain to terms in order to be able to associate a value with them.

Definition 10. Let $\mathcal{M} = \langle \mathcal{D}, I \rangle$ be a model for the language $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$, and let \mathcal{A} be an assignment in \mathcal{M} . To each $t \in \mathcal{T}_{L(\mathcal{R}, \mathcal{F}, \mathcal{C})}$ we associate a value $t^{I, \mathcal{A}}$ in \mathcal{D} as follows:

1. For $c \in \mathcal{C}$, $c^{I, \mathcal{A}} \in c^I$
2. For a variable v , $v^{I, \mathcal{A}} = v^{\mathcal{A}}$
3. For $f \in \mathcal{F}$, $[f(t_1, t_2, \dots, t_n)]^{I, \mathcal{A}} = f^I(t_1^{I, \mathcal{A}}, t_2^{I, \mathcal{A}}, \dots, t_n^{I, \mathcal{A}})$

By defining how each term is associated with a value we arrived at the point where it is possible to associate a truth value with each formula defined in $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$. This can be done in analogy to propositional logic. We are also going to use the same propositional constants \top and \perp as well as the same connectives (denoted as \circ) as in propositional logic.

Definition 11. Let $\mathcal{M} = \langle \mathcal{D}, I \rangle$ be a model for $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$. Let \mathcal{A} be an assignment in this model. To each formula Φ of $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ we associate a value $\Phi^{I, \mathcal{A}} \in \mathcal{T}$ as follows:

1. For atomic cases
 - $[P(t_1, t_2, \dots, t_n)]^{I, \mathcal{A}} = t \Leftrightarrow \langle t_1^{I, \mathcal{A}}, t_2^{I, \mathcal{A}}, \dots, t_n^{I, \mathcal{A}} \rangle \in P^I$,
 - $\top^{I, \mathcal{A}} = t$,
 - $\perp^{I, \mathcal{A}} = f$.
2. $[\neg X]^{I, \mathcal{A}} = \neg[X^{I, \mathcal{A}}]$.
3. $[X \circ Y]^{I, \mathcal{A}} = X^{I, \mathcal{A}} \circ Y^{I, \mathcal{A}}$
4. $[(\forall x)\Phi]^{I, \mathcal{A}} = t \Leftrightarrow \Phi^{I, \mathcal{B}} = t$ for every assignment \mathcal{B} in \mathcal{M} that is an x -variant of \mathcal{A} .
5. $[(\exists x)\Phi]^{I, \mathcal{A}} = t \Leftrightarrow \Phi^{I, \mathcal{B}} = t$ for some assignment \mathcal{B} in \mathcal{M} that is an x -variant of \mathcal{A} .

Here the assignment \mathcal{B} in the model \mathcal{M} is an x -variant of the assignment \mathcal{A} , if \mathcal{A} and \mathcal{B} only assign a possibly different value to x .

We want to finish this sub-section by giving an example to illustrate the expressiveness of first-order logic.

For this example we assume the language $L(\{R\}, \{\oplus\}, \emptyset)$ with variables x and y . We choose $\mathcal{M} = \langle \mathbb{N}, I \rangle$ as the model with I defined as follows:

1. $\oplus^I(a, b) = a + b$, with $a, b \in \mathbb{N}$, and
2. $R^I = \{(x, y) : x, y \in \mathbb{N}, x > y\}$

Now, consider the sentence $(\forall x)(\forall y)(\exists z)R(x \oplus y, z)$. It is easy to see that this sentence always evaluates to truth value 1 in model \mathcal{M} . Note that this valuation does not depend on the assignment as all variables in this sentence are bound to a quantifier.

As for propositional logic we could now start to introduce proof procedures which would show the full power of first-order logic. However, we skip these interesting details and refer the interested readers to [21].

4.2 Algebraic Logic

Algebraic Logic is the field of research which deals with studies of algebras that are relevant for logic. It additionally investigates the methodology of solving problems in one of the two domains, algebra or logic, and translating the solution back into its original domain.

This section is going to show how propositional logic as well as first order logic are connected to algebra.

Propositional Logic and Boolean Algebra In Sub-section 4.1 we introduced the basic syntax and semantic of propositional logic. We defined Boolean evaluation and showed how a simple formula in propositional logic is evaluated. If we want to generalise the Boolean valuation for arbitrary formula with n propositions we will have to investigate 2^n different *interpretations*.

This characteristic becomes a problem if we do not simply want to evaluate propositional formula but if we want to draw *logical consequences* from them.

Definition 12. A propositional formula X is a logical (propositional) consequence of a set S of propositional formula, provided that every model of S is also a model for X . We write $S \models_p X$

In common language we could state that a logical consequence is a statement which follows from some other statements. In mathematics, for example, the set of statements could be axioms.

So if we want to *prove* in propositional logic that a statement is a logical consequence of other propositional formula one may use established proof procedures such as Hilbert systems or resolution. Logical consequences can also be interpreted as a means to find propositional formula which are equivalent, this is, which have the same model. In the field of electronic circuits this is important as it allows to reduce cost and to find formula with the least number of connectives in it. This may be achieved by simply guessing formula and proving their logical equivalence by valuation.

In Section 2 we discussed the basic concepts of abstract algebra. We learnt that the basic language of abstract algebra is set theory and that algebra can be used to apply transformations to sets without destroying the relations between the elements within the set (homomorphism, isomorphism, etc). With this motivation we take a closer look at propositional logic.

Due to the definition of propositional logic any propositional formula has infinite equivalent formula. As an example consider the atomic formula which consists of the proposition P . This formula can easily be extended with the propositional letter \perp to $P \vee \perp$. Both formula are equivalent as they have the same model. If we now merge all propositional formula that we can build from the propositions P and R and that have the same model as P , we obtain a set that represents an equivalence class of propositional formula with model M . Obviously, we can group all propositional formula based on the propositions in $\mathcal{P} = \{P, R\}$ and obtain 16 equivalence classes. We call the set of these classes, set \mathcal{E} .

Now, according to the notion of *algebra*, we look at the interrelation between these equivalence classes. For this purpose we introduce a *tautology* and the concept of *satisfiability*.

Definition 13. A propositional formula X is a tautology if $v(X) = 1$ for every interpretation v .

Definition 14. A set S of propositional formula is satisfiable if there is some interpretation v : $v(s) = 1, \forall s \in S$.

Thus, the propositional formula $P \vee \neg P$ is a tautology. Consequently, we also find an equivalence class that only contains tautologies. As $v(\top) = 1$ for all interpretations v , we call this equivalence class

$[\top]$. Its counterpart is the equivalence class which is not satisfiable. We denote this equivalence class with $[\perp]$.

Intuitively, we now want to define a partial order on \mathcal{E} . We define the order \leq as follows:

$$X \leq Y \Leftrightarrow v(P) = v(P \wedge R), \forall P \in X \text{ and } \forall R \in Y, \text{ with } X, Y \in \mathcal{E} \text{ and for all interpretations } v.$$

Why is this an intuitive definition? Consider the valuation of formula $P \wedge R$ with the interpretation v with $v(P) = \top$ and $v(R) = \perp$. This yields $v(\top \wedge \perp) = v(\perp)$ which also means that $\perp \leq \top$ which would be an intuitive definition of a relation on the set $\{\perp, \top\}$.

We draw this partial order in Fig. 9⁶. Note that this figure is a Hasse diagram which is a graph representation for partially ordered sets (posets) (\mathcal{H}, \leq) . Vertices $a, b \in \mathcal{H}, a \neq b$ are only connected by an edge if $a \leq b$ and there is no $c \in \mathcal{H}$ such that $a \leq c \leq b$. Additionally, vertex b is drawn higher than vertex a if $a \leq b$. The use of a Hasse diagram is advantageous for our following discussions as the diagram directly shows infimum and supremum of two elements in set \mathcal{H} .

Definition 15. The infimum of a subset S of a partially ordered set (\mathcal{H}, \leq) is an element $h \in \mathcal{H}$ such that $h = \inf(S) = \max(\{x \in \mathcal{H} \mid \forall y \in S, x \leq y\})$.

Definition 16. The supremum of a subset S of a partially ordered set (\mathcal{H}, \leq) is an element $h \in \mathcal{H}$ such that $h = \sup(S) = \min(\{x \in \mathcal{H} \mid \forall y \in S, y \leq x\})$.

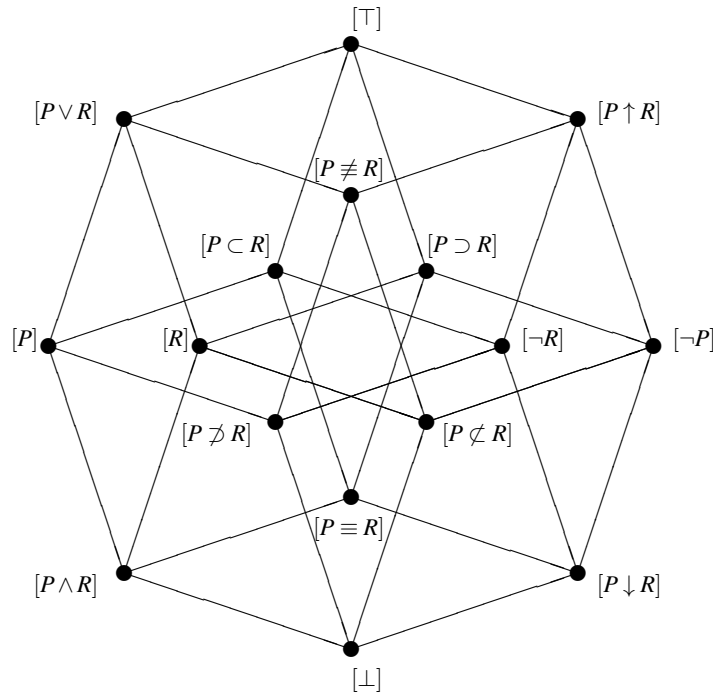


Fig. 9. Hasse diagram of a Boolean lattice of propositional logic generated by $\{P, R\}$

If we now take a closer look at this poset (\mathcal{E}, \leq) by drawing its Hasse diagram representation we will recognise two things.

First of all the poset is bounded by a least and a greatest element, \perp and \top respectively.

The second observation is not that obvious and requires a trained eye for propositional logic formula. Take representatives X and Y from two arbitrary equivalence classes $[X]$ and $[Y]$. The application of

⁶ Inspired by the slides of the talk “A Refined Geometry of Logic” by David Miller given at the Department of Mathematics, University of Warwick, Nov 2005

the propositional connectives \vee and \wedge to X and Y yields a propositional formula. This formula Z is a representative of an equivalence class $[Z]$. It is the supremum and the infimum respectively of $\{[X], [Y]\}$ and $[Z] \in \mathcal{E}$.

To illustrate this we consider the arbitrarily picked equivalence classes $[P \vee R]$ and $[\neg P]$ which form the new set $\{[P \vee R], [\neg P]\}$. We form the propositional formula $V = (P \vee R) \vee \neg P$ and $W = (P \vee R) \wedge \neg P$.

Table 3 shows the possible interpretations of formula V and W .

$v(P)$	$v(R)$	$v(P \vee R)$	$v(\neg P)$	$v(V)$	$v(W)$
0	0	0	1	1	0
0	1	1	1	1	1
1	0	1	0	1	0
1	1	1	0	1	0

Table 3. All interpretations of supremum and infimum of $v(P \vee R)$ and $v(\neg P)$

If we compare the truth values with Table 2 we can verify that V is equivalent to \top and W is equivalent to $P \not\subseteq R$. Their equivalence classes are again part of \mathcal{E} . Due to the fact that we used a Hasse diagram to display our partial order (\mathcal{E}, \leq) we can instantly see, that $[\top]$ is supremum and $[P \not\subseteq R]$ is infimum of set $\{[P \vee R], [\neg P]\}$.

Here, we already see, that the order theoretic interpretation based on the partial order \leq finds its equivalence in an algebraic structure based on the two operators \vee and \wedge and vice versa.

Based on this observation and using the above Hasse diagram we may additionally verify that any three elements of \mathcal{E} of the presented structure (\mathcal{E}, \leq) also have the characteristics of

1. associativity,
2. commutativity, and
3. absorption.

Thus the partial order (\mathcal{E}, \leq) is a lattice. Additionally, we can show the

4. distributivity of any three elements and
5. existence of the complement to each representative of an equivalence class.

As we skipped the informal proof of characteristics one through three we would like to show the distributive characteristic by example, using the Hasse diagram in figure 9.

Take three arbitrary elements from \mathcal{E} , e.g. the propositional formula $A = \neg R$, $B = \neg P$, and $C = P \not\subseteq R$. The distributivity of lattice (\mathcal{E}, \leq) or its correspondent $(\mathcal{E}, \vee, \wedge)$ requires the following equivalence

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C), \text{ or}$$

translated into order theory

$$\inf(\{A, (\sup(\{B, C\}))\}) = \sup(\{\inf(\{A, B\}), \inf(\{A, C\})\})$$

The latter can be easily verified using Hasse diagrams. In our case it is $\sup(\{B, C\}) = P \uparrow R$, $\inf(\{A, B\}) = P \downarrow R$, and $\inf(\{A, C\}) = \perp$. Thus, it is $\inf(\{A, P \uparrow R\}) = P \downarrow R$ and $\sup(\{P \downarrow R, \perp\}) = P \downarrow R$ which proves distributivity for the chosen formula.

The five characteristics listed above represent the axioms which have been set up by George Boole. Therefore, this distributive complemented lattice is also called a Boolean algebra.

Tarski and Lindenbaum were the first to precisely discuss the set of propositional formula as an algebra with operators which were induced by the connectives of the propositional language. The structural analysis we tried to sketch above by using intuition and geometrical representation in Hasse diagrams is discussed in more detail in [10,22].

We generalise this result and give the definition for a Boolean algebra.

Definition 17.

A Boolean algebra is a structure $\mathcal{B} = \langle B, +, \cdot, \neg, 0, 1 \rangle$ where the following system of equations is valid, and where $x, y, z \in B$:

$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	(associativity)
$x + y = y + x$	$x \cdot y = y \cdot x$	(commutativity)
$x + (x \cdot y) = x$	$x \cdot (x + y) = x$	(absorption)
$x + (y \cdot z) = (x + y) \cdot (x + z)$	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$	(distributivity)
$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$	(existence of complement)

With this mathematical structure the Boolean algebra for propositional logic (PL) can be defined as the Boolean algebra model $\mathcal{B}_{PL} = \langle \mathcal{E}, \vee, \wedge, \neg, [\perp], [\top] \rangle$.

You will realise that this algebra is only based on the operators \vee , \wedge , and \neg . As we have shown the equivalence with propositional logic is still valid. However, it becomes even more obvious if we reassure the reader that every propositional formula can be rewritten into an equivalent propositional formula in *normal form* which is only based on the connectives \vee and \wedge .

Summarising the results of this section we can state that with Boolean algebra we possess a very powerful tool which can be used to transform arbitrary propositional formula into other propositional and equivalent formula. These equivalence transformations have numerous areas of applications, such as integrated circuit optimisation or theorem proving to name just two possible domains.

However, of major importance for this contribution is the observation that there is a strong link between propositional logic and algebra. We saw that the definition and structure of propositional logic directly induces a Boolean algebra. Thus, we are now capable to choose a domain, propositional logic or Boolean algebra, which offers the most suitable tools and the best knowledge to analyse a structure. Consequently, it would be good if the same correspondence between first-order logic and algebra held.

First-Order Logic and Quantifier Algebras In this section we are going to show that first-order logic also possesses an algebraisation. As we can use the results from the last section we forego an intuitive interpretation and graphical explanation and use a more mathematical approach. Nevertheless, this section is going to avoid complicated and highly mathematical algebraisations of first order logic and follows the spirit of Charles C. Printer who followed “... the most satisfactory way of introducing the [...] non-specialist to the ideas and methods of algebraic logic” [23].

From Section 4.1 we already know that first-order logic is able to express formula which are not expressible in propositional logic. This is basically due to the fact that propositional logic has been extended by quantifiers and that it supports n-ary relations as opposed to strict bi-nary relations. Now, one may assume that we simply extend the Boolean algebra, the algebraisation of propositional logic, and obtain a first-order logic algebra. In the next couple of paragraphs this is exactly what we are going to do.

For this purpose we first introduce *quantifier algebras* for formula. Their definition is very similar to the construction of a Boolean algebra out of propositional logic. Thus, from the last section we simply collect the elements which we need for a formal definition.

Let Γ be the first-order language $L(\mathcal{R}, \mathcal{F}, \mathcal{C})$ and let $\langle v_{\kappa} \rangle_{\kappa < \alpha}$ a sequence of variables. Let Θ be a theory of Γ . We define \mathcal{F}^{Γ} as the set of all formula of Γ . We currently have to restrict our considerations to the set of formula which does not contain the formula $F \equiv_{\Theta} G$. Here relation \equiv_{Θ} denotes the equality relation which can be deduced from Θ . This set is denoted by $\mathcal{F}^{\Gamma} / \equiv_{\Theta}$. We will account for this restriction later in this section.

Based on $L(\mathcal{R}, \mathcal{F}^\Gamma / \equiv_\Theta, C)$ and the general Boolean algebra $\mathcal{B} = \langle B, +, \cdot, -, 0, 1 \rangle$ we can define the following Boolean operations:

$$\begin{aligned} (F / \equiv) + (G / \equiv) &= F \vee G / \equiv \\ (F / \equiv) \cdot (G / \equiv) &= F \wedge G / \equiv \\ \overline{(F / \equiv)} &= \neg F / \equiv \end{aligned}$$

1 denotes all formula of theory Θ . For simplicity and consistency with previous sections we write \top . 0, the negations of all formula in \top is denoted by \perp . We obtain the Boolean algebra $\langle \mathcal{F}^\Gamma / \equiv_\Theta, +, \cdot, -, \perp, \top \rangle$.

To define quantifier algebras we are only missing two more operations which find their analogy in the quantifiers in first-order logic. For \exists we define the operation \exists_κ with $\exists_\kappa(F / \equiv)$ denotes the equivalence class of all formula $(\exists v_\kappa)F$. Quantifier \forall can not be defined directly. Instead we define a substitution operation S_λ^κ . $S_\lambda^\kappa(F / \equiv)$ denotes the equivalence class of the formula which results from F by replacing each free occurrence of v_κ by v_λ . Here it becomes obvious why we needed an ordered sequence of variables. Extending the Boolean algebra above we obtain the *quantifier algebra of formula*: $\langle \mathcal{F}^\Gamma / \equiv_\Theta, +, \cdot, -, \perp, \top, S_\lambda^\kappa, \exists_\kappa \rangle_{\kappa, \lambda < \alpha}$ associated with Θ . We now formally define this algebra:

Definition 18. By a *quantifier algebra of degree α (QA_α)* we mean a structure

$\mathcal{U} = \langle A, +, \cdot, -, 0, 1, S_\lambda^\kappa, \exists_\kappa \rangle_{\kappa, \lambda < \alpha}$ where $\langle A, +, \cdot, -, 0, 1 \rangle$ is a Boolean Algebra with the unary operators S_λ^κ and \exists_κ which have the following properties for all $x, y \in A$ and $\kappa, \gamma, \lambda < \alpha, \alpha \geq 2$:

$$\begin{aligned} (q_1) \quad S_\lambda^\kappa(\bar{x}) &= \overline{S_\lambda^\kappa(x)} \\ (q_2) \quad S_\lambda^\kappa(x+y) &= S_\lambda^\kappa(x) + S_\lambda^\kappa(y) \\ (q_3) \quad S_\kappa^\kappa(x) &= x \\ (q_4) \quad S_\lambda^\kappa S_\kappa^\gamma & \\ (q_5) \quad \exists_\kappa(x+y) &= \exists_\kappa x + \exists_\kappa y \\ (q_6) \quad x &\leq \exists_\kappa x \\ (q_7) \quad S_\lambda^\kappa \exists_\kappa &= \exists_\kappa \\ (q_8) \quad \exists_\kappa S_\lambda^\kappa &= S_\lambda^\kappa \text{ if } \kappa \neq \lambda \\ (q_9) \quad S_\lambda^\kappa \exists_\gamma &= \exists_\gamma S_\lambda^\kappa \text{ if } \gamma \neq \kappa, \lambda \end{aligned}$$

Due to the many indexes these equations may at first look difficult but if you have a closer look you will realise that you can group $(q_1) - (q_4)$ into simple substitution properties, $(q_5) - (q_6)$ into quantifier properties, and $(q_7) - (q_9)$ into a group which relate substitutions to quantifiers. We assume here that the interpretation of these equations is obvious.

If \mathcal{U} is a quantifier algebra as defined above we can define a so called *dimension set* of a formula $a \in A$:

$$\diamond x = \{ \kappa < \alpha : \forall \lambda \neq \kappa, S_\lambda^\kappa x \neq x \}$$

Quantitatively this is the set of all indexes κ of variables v_κ which would change the valuation of formula x if substituted with another variable v_λ . We define \mathcal{U} to be *locally finite* if $\diamond x$ is a finite set.

It can be shown that every quantifier algebra *of formula* is a locally finite quantifier algebra as defined above. Accordingly, one can show that if \mathcal{U} is a locally finite quantifier algebra then there is a theory Θ such that \mathcal{U} is isomorphic to a quantifier algebra of formula which could be derived from Θ .

This result is already very important as it implies that we can express every theory in first-order logic without equality by using locally finite quantifier algebras and thus sets up a link between algebra and logic. Conversely we can take a locally finite quantifier algebra and translate it into a first-order logic without equality.

To make this link even stronger we will need to remove the limitations from above which restricted our first-order formula to $\mathcal{F}^\Gamma / \equiv_\Theta$.

Intuitively we will extend the quantifier algebras by another equivalence class. This is equivalent to extending Boolean algebra with substitution and existence equivalence classes. We define the equivalence class of equality as $e_{\kappa\lambda}$ which contains all formula $v_{\kappa} \equiv v_{\lambda}$. Finally, we define the quantifier algebra with equality.

Definition 19. A quantifier algebra with equality is an algebra $\langle A, +, \cdot, -, 0, 1, S_{\lambda}^{\kappa}, \exists_{\kappa}, e_{\kappa\lambda} \rangle_{\kappa, \lambda < \alpha}$ such that $\mathcal{U} = \langle A, +, \cdot, -, 0, 1, S_{\lambda}^{\kappa}, \exists_{\kappa} \rangle_{\kappa, \lambda < \alpha}$ is a QA_{α} and $e_{\kappa\lambda}$ are distinguished elements which satisfy

$$(q_{10}) \quad S_{\lambda}^{\kappa} e_{\kappa\lambda} = 1$$

$$(q_{11}) \quad x \cdot e_{\kappa\lambda} \leq S_{\lambda}^{\kappa} x$$

Why can we be sure that this quantifier algebra with equality is finally an algebra which represents our first-order logic? Common practice is to search for an algebra from which we know that it is an algebraisation of first-order logic and show that quantifier algebra with equality is isomorphic to this algebra.

Henkin, Monk and Tarski defined so called *cylindric algebras* in [11] as follows.

Definition 20. By a cylindric algebra of degree α we mean a system $\langle A, +, \cdot, -, 0, 1, \exists_{\kappa}, e_{\kappa\lambda} \rangle_{\kappa, \lambda < \alpha}$ such that $\langle A, +, \cdot, -, 0, 1 \rangle$ is a Boolean algebra and \exists_{κ} and $e_{\kappa\lambda}$ satisfy the following conditions $\forall x \in A$ and $\kappa, \lambda < \alpha$:

$$(c_1) \quad \exists_{\kappa} 0 = 0$$

$$(c_2) \quad x \leq \exists_{\kappa} x$$

$$(c_3) \quad \exists_{\kappa}(x \cdot \exists_{\kappa} y) = \exists_{\kappa} x \cdot \exists_{\kappa} y$$

$$(c_4) \quad \exists_{\kappa} \exists_{\lambda} = \exists_{\lambda} \exists_{\kappa}$$

$$(c_5) \quad e_{\kappa\kappa} = 1$$

$$(c_6) \quad e_{\lambda\gamma} = \exists_{\kappa}(e_{\lambda\kappa} \cdot e_{\kappa\gamma}) \text{ if } \kappa \neq \gamma, \lambda$$

$$(c_7) \quad \exists_{\kappa}(e_{\kappa\lambda} \cdot x) \cdot \exists_{\kappa}(e_{\kappa\lambda} \cdot \bar{x}) = 0 \text{ if } \kappa \neq \lambda$$

You may already wonder why we define this algebra. If we look at the formal definition of cylindric algebras we will recognise that the substitution operation has disappeared which accounted for the \forall quantifier in quantifier algebras with equation. In fact if operations S_{λ}^{κ} are defined as

$$S_{\lambda}^{\kappa} x = x \text{ if } \kappa = \lambda; S_{\lambda}^{\kappa} x = \exists_{\kappa}(x \cdot e_{\kappa\lambda}) \text{ if } \kappa \neq \lambda$$

This definition appears to be obvious. Of course, a formula which replaces a variable by the same variable is equivalent to its original version. Substitution with a different value can be achieved by assigning v_{λ} to v_{κ} if v_{κ} actually exists.

With this definition we can conclude that a cylindric algebra is a quantifier algebra with a equality. From [23] we additionally know that definitions $(q_1) - (q_{11})$ are actually all theorems of the theory of cylindric algebras as shown in [11].

It turns out that we now have a full mapping from cylindric algebra to first-order logic with equality. If we only consider first-order logic without equality Galler also shows in [24] that we can map quantifier algebra to polyadic algebra by slightly modifying substitution and existence operators in QA_{α} . By doing this we obtain a so called polyadic algebra defined by Halmos in [25].

As in Section 4.2 we now have the evidence that we can directly map first-order logic into algebra and vice versa. Daigneault's interpretation [26] of Krasner's general theory on Galois Fields [27] (see also Section 2.5) as polyadic algebras gives even more evidence to the relation of first-order logic and algebra.

This implies again that we can analyse effects in the domains which we can map to algebra in powerful first-order logic. One of the probably most exciting results is the relatively recent work of Andr eka, Madara asz and N emeti which used the relation between cylindric algebra and first-order logic to formulate Einstein's general theory of relativity in first-order logic [28].

Finally we want to refer the interested reader to the survey "Tarskian Algebraic Logic" by T.S. Ahmed who gives a good overview on algebraic logic, summarising its history and its different relations to other fields. This survey also contains a nice discussion of (n-ary) Cylindric Algebras and their mapping to

first-order logic. Andr eka et. al. focus in [29] more on the Tarskian structuralist view to logic and thus can also be considered to be a good complement to this section. In this work the relevant branch of *universal algebra* is also discussed. As a matter of fact universal algebra offers a framework which provides powerful tools and theories to investigate the interconnections between different classes of algebras.

4.3 Temporal Logic

Sections 4.1 and 4.1 quantitatively introduced propositional and first-order logic. Generally speaking these logics support the reasoning based on propositions or terms and formula. The truth values are fixed and constant over time, this is, no matter when you evaluate a proposition or a first-order formula, the truth value will always be the same only depending on the valuation function, the propositions, and variables used.

Temporal logic extends the classical concept and introduces the dimension of time. Thus, this notion of logic will extend propositions with a reference to time conditions. Consequently, compared with classical logic which can describe states and properties of systems, temporal logic is able to express sequences of state changes and properties of behaviour.

As we have seen in classical logic, also temporal logic comprises different logics. Thus, propositional and first-order logic find their correspondence in propositional and first-order temporal logic.

To introduce the general idea of temporal logic we will briefly introduce linear temporal logic (LTL). As we will not perform a similar algebraisation as for propositional and first-order logic, we only give an informal definition for temporal logic and briefly compare it and its variants to first-order logic. Based on these informal definitions we outline the link between temporal logic and algebras. For this purpose we also establish a link to universal algebras. Finally, this section will explain the differences between linear and branching time logic and conclude with a short list of applications of temporal logics.

Linear Time Logic For this purpose we first define two new temporal operators on a set \mathcal{P} of regular propositional formula.

1. $\bigcirc Q$ is a linear temporal formula if $Q \in \mathcal{P}$
2. $Q \cup R$ is a linear temporal formula if $Q, R \in \mathcal{P}$

To give the symbols defined above some semantics we extend the regular valuation function from Section 4.1 as follows.

1. $v(\bigcirc Q) = 1$, if and only if in the next time step $v(Q) = 1$.
2. $v(Q \cup R) = 1$, if and only if $v(Q \wedge \neg R) = 1$ until $v(R) = 1$.

To explain these rather abstract definitions we illustrate them in Fig. 10. Arrows in this figure represent the time line. Single nodes represent points in time at which a proposition changes. Above each node you find the valuation of the corresponding formula depending on the time they are evaluated.

Clearly, the choice of the type of illustration in Fig. 10 was not arbitrary. By choosing a representation which resembles finite state machines we also wanted to emphasise the fact that Pnueli saw linear temporal logic as a tools to analyse computer programs [30].

The first linear sequence in this figure presents the semantics of a simple propositional formula Q . Its valuation is given for a specific point in time (here the first time step in the figure). If time is proceeding the valuation of this formula is arbitrary. For the second operation the formula $Q \vee \neg Q$ has to evaluate to be true in the system. If this is true *until* $v(R) = 1$ the formula $Q \cup R$ evaluates true for this sequence. The last formula $\bigcirc Q$ represents the simplest sequence as it only requires Q to evaluate to $v(Q) = 1$ in the next time step of this sequence.

With the given operators of propositional logic extended by \cup (until) and \bigcirc (next) and their semantics it is clear there are other temporal operators which can be derived. We give some example in the following list:

1. $\diamond Q \equiv \top \cup Q$, with $Q \in \mathcal{P}$

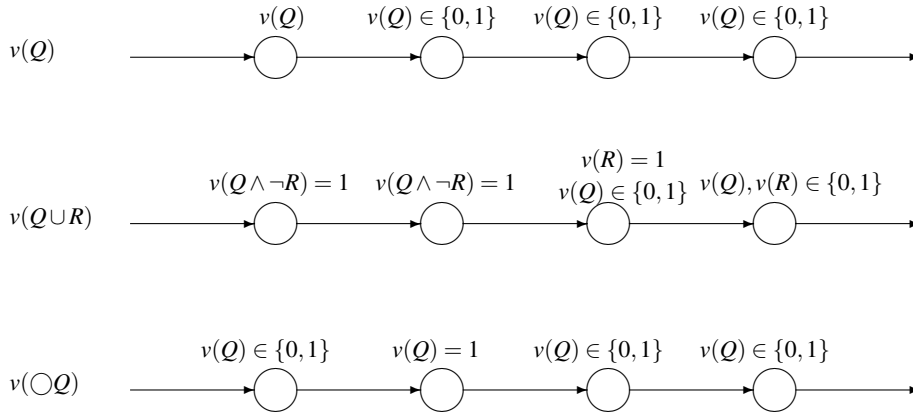


Fig. 10. Illustration of the semantics of LTL operators

2. $\Box Q \equiv \neg \Diamond \neg Q$, with $Q \in \mathcal{P}$

The semantics of these operators is already defined through the equivalence with formula that use the operators we defined above. To clarify their meaning we again use the same illustration as in figure 10.

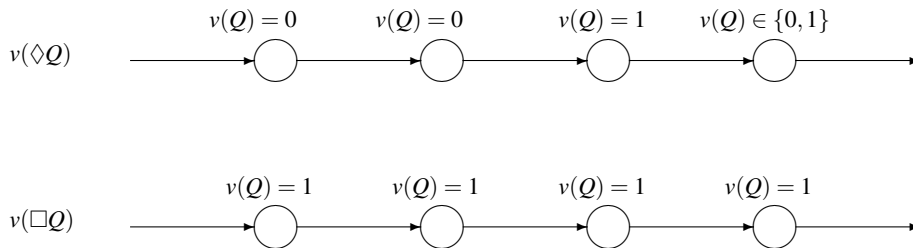


Fig. 11. Illustration of the semantics of derived LTL operators

As \Box depends on \Diamond we start to look at the representation of $\Diamond Q$ as illustrated in Fig. 11. Unfortunately this representation does not show that the valuation of Q does not have to turn to true after exactly three time steps. This can happen at any time such that $v(\Diamond Q) = 1$ if Q eventually evaluates to 1. If we look at the semantics of \cup this behaviour becomes even more clear as the condition which has to hold has to hold before Q can come true is always true as it is the tautology \top .

At the bottom of Fig. 11 the operator \Box is illustrated. It turns out that this operator requires the formula it is associated to. With this knowledge we look back to our definition of $\Box Q$ and first consider the formula $\Box \neg Q$. As we learnt above this states that $\neg Q$ will eventually evaluate to true, this is, eventually Q evaluates to false. So if we negate this statement, as done in the definition above, we obtain that Q will not evaluate to false eventually which is equivalent to: “Proposition Q is *always* true”.

To show the power of this new approach we may use an example which is used in many lectures. Take a traffic light. With propositional logic you will be able to set up propositions about properties of its static characteristics, such as that the green and red light are not illuminated at the same time. Here you can make statements about time because this characteristic is an invariant. However, you will not be able to describe that if the traffic light shows red light it will eventually turn into red light. In LTL you can express this state change with the formula $\Box(R \supset \Diamond G)$. Here R denotes the proposition “Red light” and G the proposition “Green light”. Reading this formula in natural language yields: It is *always* true that if there is red light then *eventually* there will be green light.

Let us now look at the structure of linear temporal logic. To define it we took a classical propositional logic and extended it with two additional operators which introduced the time dimension. However, if we look at the definition of these operators then we could always formulate them with *there exists* a point in time or *for all* points in time. Thus, these definitions suggest themselves to ask whether we could model linear temporal logic in first order logic.

In fact, it is possible but tedious because we have to model time in first-order logic and thus formula become very complex and difficult to read. This is comparable to translating a higher programming language to assembler. However, the important result we should remember is that we can model all statements in propositional temporal logic using first-order logic. In [31] Etesami et al. go even one step further and prove that unary temporal logic (which is temporal logic with only unary temporal operators as defined above) can be expressed by first-order logic with only two variables. Consequently, we can use the algebras developed in the last section to allow the algebraisation of unary-temporal logic.

One obvious question follows: Can formula in first-order temporal logic be translated into first-order logic and thus is it possible to use the same algebras? The answer to this question would require the coverage of more theoretical concepts and we would need to extend our discussion of logic to completeness and other important theories. Therefore, for completeness we mention here that a lot of research has been conducted in this area. A good overview and introduction can be found in [32] and [33].

Branching Time Logic So far linear temporal logics have been discussed. They get their name from the fact that they consider only behaviours which can be modelled as linear time sequences. This characteristic is nicely illustrated by Fig. 10 and 11. Every state, represented as a node, has exactly one successor. However, in communication systems or generally in concurrent systems a state in time will need to have several future states. To model such system, branching time logics [34] have been proposed. They possess a tree structure in which each state in time has more than one successor. One of the most popular of these logics is the computation tree logic (CTL) proposed in [35].

It is usually easier to model concurrent systems using branching time logic. This is due to the fact that their additional path quantifiers usually support the navigation in their tree structure. However, sometimes it is easier to use existing tools, proof techniques, and analytical methods which exist already in one domain. We can state here that branching temporal logics can be translated into linear temporal logic by simply modelling each branch in the branching logic as a linear sequence in linear temporal logic. This is common practice when, for example, translating non-deterministic machine models into deterministic ones.

Applications In this section we have seen how temporal logic can extend classical propositional logic to describe properties of behaviour or generally time-dependent system characteristics. This expressiveness can be used in many different ways. Very popular is the use of various types of temporal logic in the field of security. Here the logical formula are used to describe expected behaviour. This behaviour could reflect certain security characteristics of a control flow of a computer program, of a security protocol, or a general access control mechanism. Accordingly, the number of application of temporal logic in the field of security is huge.

The following list shows only an overview of areas in which temporal logic is and can be applied and emphasises its relevance to security.

Formal Specification has currently a strong focus in many areas of security research. Distributed computing systems, access control and software systems, security protocols, etc. may be subject to using logic for specifying their security characteristics [36]. This notion of formal verification was mainly induced by Pnueli [30,37] and Lamport [38].

Formal Verification One important characteristic of classical and temporal logic is the existence of a proof calculus which is mainly based on the mathematical foundation of these logics in algebra. This calculus can be used to show the correctness of system specifications based on logics [37,38].

Requirements Description Specifying the security-compliant operation of a system does not only require the thorough specification of its components. It is also required to thoroughly specify how the

system is restricted and what the environment can or cannot do. Formula of (temporal) logic can be used to specify these requirements.

5 Conclusion

The reason we are interested in the metabolism of the cell is that the cell can be considered an immensely complex parallel computer that executes a ‘distributed algorithm’. This term arises from the fact that even though most of the instructions are coded in the DNA, a significant part of each metabolic cycle depends on the chemical composition of the cell moment-by-moment. The DNA instructions are propagated through the cell by diffusion mechanisms coupled with various reactions. The concentrations of the various chemical species are far from uniform. In addition, several kinds of membranes and structural elements separate areas of different chemical activities and make the internal topology of the cell nested and extremely complex. There is however an aspect that greatly simplifies the conceptualisation of internal cell operations: dimensional reduction.

The most successful example of dimensional reduction is provided by the microcanonical ensemble of equilibrium statistical mechanics: an isolated system will approach equilibrium, which corresponds to the configuration of highest entropy. The configuration of highest entropy is, by definition, the most probable. Thus it can be easily identified by the peak in the frequency distribution of all possible configurations. For an isolated system in equilibrium this is nothing more than the familiar Gaussian, which has a very sharp peak indeed since it decays on both sides of the maximum like a square exponential. More generally, the Central Limit Theorem says that a sequence of random samples will converge to a Gaussian for equilibrium systems [39]. A more detailed discussion of these physics concepts and their relationship to self-organisation can be found in [4]. For our purposes here it is sufficient to recall how the CLT allows the derivation of stable macroscopic properties such as pressure from the random collisions of $O(10^{23})$ molecules in a litre of air. That’s a dimensional reduction of 10^{23} to 1.

When a gene is activated and begins to signal to the cell machinery to fabricate a particular protein, it creates several thousand mRNA molecules that set an equal number of ribosomes to work in the cytoplasm (each cell has millions of ribosomes, or ‘protein factories’). Such a large number of proteins will provide a high probability that the particular function the gene wants to execute will be executed. Therefore, we can regard the large numbers of molecules in the cell as a strategy to achieve a form of dimensional reduction that in computer science we generally call ‘abstraction’. Several thousand proteins will participate in a relatively few biochemical reactions to advance one or more metabolic cycles one execution step. Even though the interior of the cell is never in equilibrium (it relies on its ‘fall’ toward equilibrium as the engine that drives all of its spontaneous self-organising processes—in fact, that’s what ‘spontaneous’ means), its complex topology is divided into many areas in each of which a few reactions are active at any one point in time. From millions of elements we can therefore see how through a relatively small number of quasi-equilibrium regions of the cell several hundred metabolic cycles can be executed in parallel.

Dimensional reduction or abstraction working together with the fact that the DNA itself is composed of genes that can be ON or OFF makes it sound plausible that the internal working of the cell can be modelled through a discrete or digital framework. We can begin to recognise some of the concepts discussed in Section 2. For example, the 4 DNA bases represent an alphabet with which the specification of proteins can be coded. The architecture of the DNA is such that it must not only carry the genetic code but also support its expression through interactions with its environment. This has been achieved by replicating the same information along the two parallel strands of the DNA molecule, in such a way that the 4 bases are paired up two-by-two. In other words, of the four bases Thymine (T), Cytosine (C), Adenine (A), and Guanine (G) [40], only 2 kinds of pairs are possible: A-T and C-G, so that a binary base field might still be relevant in some way.

Assuming we can recognise an algebraic structure in the digital nature of cell biology, the same or similar structure would help us make a bridge to the structure of logic. So at this point we may ask the question: What can we do with logic and biology? As we have explained in Section 4.2 it may be possible that by studying biological systems our research may yield interesting connections between two, at first sight, completely different domains. However, this is not the final goal of our work.

In BIONETS we are currently exploring characteristics and structure of and operations on Fraglets [41]. They represent a programming model which is based on multi-set rewriting and can be compared to the copying of DNA sequences described above. In fact, it turns out that the execution model of Fraglets is very similar to interacting biological systems such as the DNA with the various enzymes that it generates. As a consequence, one may actually expect that the implementation of genetic algorithms on these structures may be fairly easy. However, first experiments show that this is not the case. To successfully design a system which performs genetic operations we need to know what the basic building blocks are with which we may build new individuals, and which structures of the individual representation (in our case a program or service) are relevant to yield a 'fit' individual. We know that biology is very successful in choosing the right building blocks and the correct genetic operations that are applied to the appropriate portions of the DNA. By choosing a programming model which is very similar to bio-chemical processes in a cell we hope to be able to transfer the observations made in the realm of biology to a programming language. In abstract terms, we would like to be able to describe structures of Fraglets using algebra. These structures and their interaction with the environment would correspond to specific program characteristics and behaviour. This seems possible in principle because we would exploit our observations from biological system. Here this process is comparable to specific structures of the DNA (genes) which in interaction with their environment yield different phenotypes of an organism. In this way, Fraglets and their similarity to bio-chemical processes form the exemplary bridge head of the application of our theory to security.

As we learnt in this document we can use an algebra and map it into the realm of logic. Depending on the type of algebra we obtain from the analysis of our programming model we will have the possibility to analyse the corresponding characteristics in the realm of logic. As explained in Section 4.3 logic is a powerful means to investigate program properties, including security. Clearly, this process has its limitations as we will not be able to investigate any arbitrary program characteristic (for example the Halting problem). However, it will be an important step towards understanding the complicated programming and execution model of Fraglets and possibly their counterpart in biology, the DNA and its proteins. Furthermore, if we invert this analytical process, it becomes clear how we could guide evolution. Being able to express specific program properties in logic, we will be able to express the same characteristic in algebra and thus as a structure of the programming language, the Fraglets.

Consequently, the insights that we hope to obtain from this bridge between biology and logic could also help to improve genetic operators. This is due to the fact that it would become easier to investigate the implications of a structural change (which corresponds to a genetic operation) on the program properties. Finally, this would also have immediate effect on the design of fitness functions. Based on the knowledge of which structure of the program representation is responsible for which program property, fitness functions could be improved. Instead of evaluating program representations as a whole they could analyse their structure and evaluate only those parts that are relevant for "survival". This will also enable better evaluation of security characteristics of the evolved programs.

The final goal is to be able to specify the security or other functional characteristics of a digital system, and have the specifications map to running code through a process analogous to gene expression, i.e. constructing order through interaction with the environment.

Acknowledgements

The authors would like to acknowledge the valuable feedback they received from Luciana Pelusi and Marinella Petrocchi on the first draft of this article, and the stimulating conversations they had with Sotiris Moschoyiannis and Christopher Alm during the course of the research.

The first author would like to acknowledge the partial support of his work by the OPAALS Project, contract number FP6-034824.

References

1. S. Kauffman, *The Origins of Order: Self-Organisation and Selection in Evolution*. Oxford: Oxford University Press, 1993.

2. H. Maturana and F. Varela, *The Tree of Knowledge. The Biological Roots of Human Understanding*. Boston and London: Shambhala, 1998.
3. P. Dini, *D4.1-DBE Science Vision*. www.digital-ecosystem.org, Deliverables: SP0: DBE Project, 2006.
4. P. Dini and E. Berdou, *D18.1-Report on DBE-Specific Use Cases*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2004.
5. R. Sanchez, E. Morgado, and R. Grau, "Gene algebra from a genetic code algebraic structure," *Journal of Mathematical Biology*, vol. 51, pp. 431–457, 2005.
6. P. Dini, *D18.4-Report on self-organisation from a dynamical systems and computer science viewpoint*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2007.
7. —, *D18.6-5-Year living roadmap for Digital Ecosystems research in biology-inspired computing*. www.digital-ecosystem.org, Deliverables: SP5: DBE Project, 2007.
8. G. Boole, *The Mathematical Analysis of Logic*. Bristol: Thoemmes Press, 1998 (1847).
9. P. R. Halmos, *Algebraic Logic*. AMS, 2007 (1962).
10. A. Tarski, *Logic, Semantics, Metamathematics*. Oxford University Press, 1956, edited by J. H. Woodger.
11. L. Henkin, J. D. Monk, and A. Tarski, *Cylindric Algebras*. Amsterdam: North-Holland, 1971.
12. P. Dini, G. Briscoe, A. J. Munro, and S. Lain, *DI.1: Towards a Biological and Mathematical Framework for Interaction Computing*. http://files.opaals.eu/OPAALS/Year_2_Deliverables/WP01/: OPAALS Deliverable, European Commission, 2008.
13. P. Cameron, *Introduction to Algebra*. Oxford: Oxford University Press, 1998.
14. R. B. J. T. Allenby, *Rings, Fields and Groups: An Introduction to Abstract Algebra (2nd Ed.)*. Oxford: Butterworth-Heinemann, 1991.
15. I. Stewart, *Galois Theory (2nd Ed.)*. London: Chapman and Hall, 1989.
16. J. A. Gordon, "Very Simple Method to Find the Minimum Polynomial of an Arbitrary Non-Zero Element of a Finite Field," *Electronics Letters*, vol. 12, pp. 663–664, 1976.
17. S. Moschoyiannis, *Group Theory and Error Detecting/Correcting Codes*. Guildford, UK: University of Surrey, Department of Computing, Technical Report SCOMP-TC-02-01, 2001.
18. M. Purser, *Introduction to Error Correcting Codes*. Boston: Artech House, 1995.
19. K. Erdmann and M. J. Wildon, *Introduction to Lie Algebras*. London: Springer, 2006.
20. R. Sanchez, R. Grau, and E. Morgado, "A novel Lie algebra of the genetic code over the Galois field of four DNA bases," *Mathematical Biosciences*, vol. 202, pp. 156–174, 2006.
21. M. Fitting, *First-order logic and automated theorem proving (2nd ed.)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.
22. P. Halmos and S. Givant, *Logic as Algebra*. Washington, DC: The Mathematical Association of America, 1998, vol. 21 of Dolciani Mathematical Expositions.
23. C. C. Pinter, "A Simple Algebra of First Order Logic," *Notre Dame Journal of Formal Logic*, vol. XIV, no. 3, July 1973.
24. B. A. Galler, "Cylindric and polyadic algebras," in *Proceedings of the American Mathematical Society*, vol. 8, no. 1, 1957, pp. 176–183.
25. P. R. Halmos, "Algebraic logic, i. monadic boolean algebras," *Composito Mathematica*, vol. 12, pp. 217–249, 1956.
26. A. Daigneault, "On automorphisms of polyadic algebras," *Transactions of the American Mathematical Society*, vol. 112, no. 1, pp. 84–130, 1964.
27. M. Krasner, "Généralisation abstraite de la théorie de Galois," in *Algèbre et théorie des nombres*, ser. C.N.R.S. Paris: Centre national de la Recherche scientifique, 1950, no. 24, pp. 163–168.
28. H. Andréka, I. Németi, and J. Madarász, "On the logical structure of relativity theories." Electronically available at <http://www.math-inst.hu/pub/algebraic-logic/Contents.html>, Tech. Rep., 2002.
29. H. Andréka, J. X. Madarász, and I. Németi, "Algebras of relations of various ranks, some current trends and applications," *JoRMiCS*, vol. 1, pp. 27–49, 2004.
30. A. Pnueli, "The temporal logic of programs," in *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, 1977, pp. 46–57.
31. K. Etessami, M. Y. Vardi, and T. Wilke, "First-order logic with two variables and unary temporal logic," *Information and Computation*, vol. 179, no. 2, pp. 279–295, December 2002. [Online]. Available: <http://dx.doi.org/10.1006/inco.2001.2953>
32. M. Gehrke and Y. Venema, *Algebraic Tools for Modal Logic*. Electronically available at <http://www.helsinki.fi/esslli/courses/readers/K15.pdf>, Helsinki, Finland: European Summer School in Logic, Language and Information, 2001.
33. P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*. Cambridge University Press, 2001.
34. M. Ben-Ari, Z. Manna, and A. Pnueli, "The temporal logic of branching time," in *POPL '81: Proceedings of the 8th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. New York, NY, USA: ACM Press, 1981, pp. 164–176.
35. E. M. Clarke, E. A. Emerson, and A. P. Sistla, "Automatic verification of finite state concurrent system using temporal logic specifications: a practical approach," in *POPL '83: Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. New York, NY, USA: ACM Press, 1983, pp. 117–126.
36. F. Kröger, *Temporal logic of programs*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.
37. Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems*. New York, NY, USA: Springer-Verlag New York, Inc., 1992.
38. L. Lamport, "The Temporal Logic of Actions," *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 3, pp. 872–923, May 1994. [Online]. Available: citeseer.ist.psu.edu/lamport94temporal.html

39. D. Sornette, *Critical Phenomena in Natural Sciences: Chaos, Fractals, Self-organization and Disorder - Concepts and Tools*. Heidelberg: Springer, 2000.
40. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell (4th Ed.)*. New York: Garland Science, 2002.
41. C. Tschudin, "Fraglets - a metabolistic execution model for communication protocols," in *2nd Annual Symposium on Autonomous Intelligent Networks and Systems (AINS)*, Menlo Park, USA, July 2003.

Algebraic and Categorical Framework for Interaction Computing and Symbiotic Security

Paolo Dini¹, Daniel Schreckling² and Gábor Horváth¹

¹ Department of Media and Communications
London School of Economics and Political Science
London, United Kingdom

p.dini@lse.ac.uk, g.horvath@lse.ac.uk

² Institute of IT-Security and Security Law
University of Passau
Passau, Germany

ds@sec.uni-passau.de

Abstract. This chapter builds on the previous [1] and continues the exploration and development of a theoretical and mathematical framework for biologically-inspired interaction computing, with security applications in mind. The chapter draws on a series of four BIONETS reports and summarises the research work performed in the area of the mathematics of interaction computing during the last two years of this project. The chapter provides a broad conceptual discussion of the foundations and rationale for a theory of interaction computing and for the mathematical perspectives we advocate for its development. It then provides an exhaustive discussion of a permutation group example, in a tutorial style, to provide an intuitive basis for understanding the role of transformation semigroups as a basis of an algebraic theory of computation. The formalism of category theory is then presented in relation to the specification of automata behaviour and to its relationship to automata structure. The paper ends with a synthesis of the main insights gained to date in the emerging theory of interaction computing.

1 Introduction

This chapter continues the construction of a theory of bio-computing that was begun in [1], and builds on an additional two years of research. Although the problem of interaction computing is not ‘solved’, yet, several of the tentative hypotheses we made three years ago appear more plausible now, and are backed up by a significantly wider and deeper body of theory. Although our research has not reached any major new breakthroughs, e.g. in the form of new theorems, a usable formalisation, or an implementable architecture, we think that the framework we have developed has brought together theories – and researchers – from different disciplines in a way that is beginning to make a consistent formalism appear possible across cell biology, algebraic automata theory, and formal languages, and that will continue to motivate our collaboration in the coming years. The framework that is beginning to emerge and the challenges that remain to be overcome seem to point to a potentially new and transformative theory of computing with wide applicability in computer science and biology.

The lack of a clear formalism for bio-computing is a consequence of the lack of a complete mathematical theory of cell biology – and, arguably, of computer science. Therefore, the first part of this chapter is mainly conceptual and documents the process by which we have narrowed down the areas of mathematics that we feel are most relevant to the problem at hand. We have cast the net wide and across several disciplines, thus the chapter relies on a correspondingly long discussion before reaching the relatively narrow focus of semigroup theory and category theory as the current focus of research. As we reach actionable conclusions, we will shift the focus to other relevant areas such as formal specification languages. Ultimately, we expect the formalism we are developing to enable a continuous flow of usable biologically-inspired models based on a systematic analysis and characterisation of cellular processes into computer science.

The emerging character of the theory, combined with the assumption of a widely interdisciplinary audience, suggested an ‘iterative’ style of presentation which, in turn, led to a rather long chapter. Thus, the

chapter is organised in sections that discuss and present the theoretical and mathematical ideas at increasing levels of depth and detail. Anyone should be able to read and understand the first sections, getting the overall ideas and concepts at a high level; the later sections become progressively more technical and mathematical, hoping that this will be found useful by theoretical computer scientists and mathematicians interested in these kinds of applications. Although the theory is not complete, this manner of presentation should make it possible for most readers to assess the plausibility of the framework we have put together and of the next steps this framework calls for.

Our collaboration began four years ago, at the beginning of the BIONETS project, with the challenge of including a security perspective within the theoretical work on the fundamental paradigms of biologically-inspired computing. We proposed the concept of ‘symbiotic security’, which captures the balance between the *encapsulation* of a security function in a specialised security service with the *interdependence* between such a service and the service(s) it is meant to secure. In other words, in order to address the problem posed by the ‘*ex post* patching paradigm’ in the development of security services and applications,³ we felt it would be worth investigating a mode of generation of such paired services based on the integration of their most fundamental functions *ex ante*, similarly to how the human immune system is inextricably integrated with the nervous and endocrine systems [2].

The above perspective on security motivated an in-depth investigation into how biological structure and behaviour could be understood from the point of view of computer science and could lead to new forms of computing. The work that was performed in this direction within the BIONETS project is summarised schematically in Figure 1. This chapter addresses the topics shown in red font and connected by red arrow in this figure. Before we can begin discussing these topics we need to summarise the progress we have made since [1] in conceptualising interaction computing.

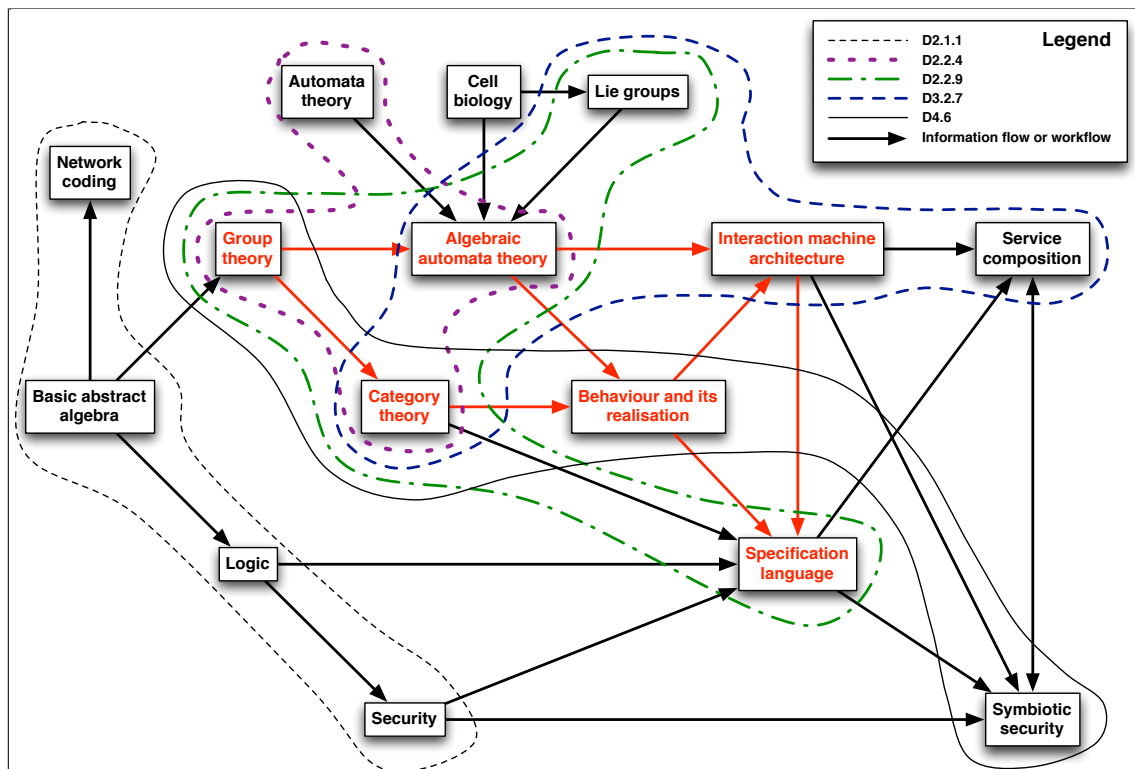


Fig. 1. Map of research topics relevant to interaction computing and symbiotic security, showing the topics discussed in various BIONETS deliverables upon which this chapter depends and, in red, topics addressed in this chapter. D2.1.1: [3] but also [1], D2.2.4: [4], D2.2.9: [5], D3.2.7: [6], D4.6: [7]

³ The *ex post* patching paradigm does not develop new security services or applications but, rather, patches the corresponding applications after realising that they have a problem.

2 High-Level Framework, Context and Motivation

2.1 High-Level Framework

Fig. 2 gives a high-level view of the theoretical research framework that will be discussed and justified in more detail in the rest of this chapter. The most important aspect of the theory that is emerging is that it needs to address three fundamental aspects of biology: structure, function, and organisation. Our preliminary results and insights point to algebra, dynamical systems, and autopoiesis, respectively, as the theories that can explain, describe, and/or model these aspects of biology and that need to be unified by a common mathematical framework that can effect a mapping to computer science. The target of these mappings appears to be a unification of the algebraic and algorithmic structure of automata, and novel ideas in software architectures and biological design patterns inspired by autopoiesis. Category theory is then able to relate any of the structures thus defined that have algebraic character to automata behaviour (which is also some kind of algebra) and from the mathematical formalisation of automata behaviour into a behaviour specification language. Instantiation of this framework in modern distributed and web-oriented computing environments may be expressible compatibly with the Representational State Transfer (REST) architectural style [8]. It is important to emphasise that the term “structure” is quite overloaded in our work. It can refer to biological (physical) structure or to algebraic structure. Hopefully the different meanings will be clear from the context.

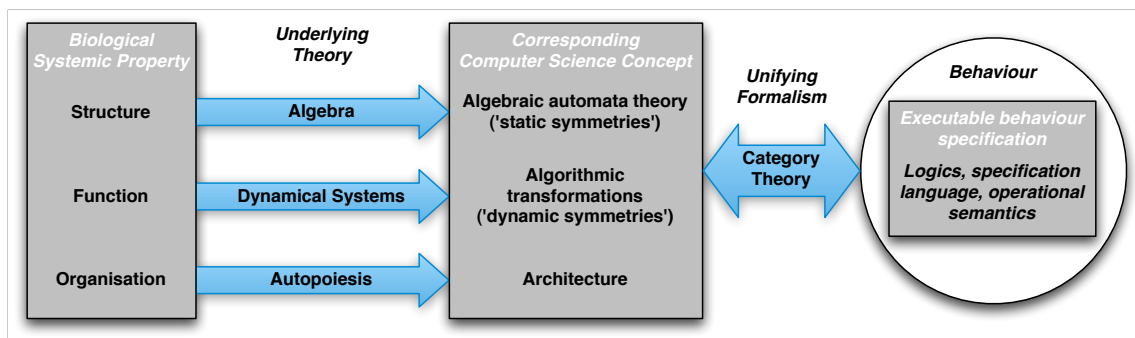


Fig. 2. High-level view of the theoretical research framework

2.2 Historical Recap

The complexity and interconnections of the research activities that are gradually unfolding in the two projects make it necessary to provide a summary of past activities and to retrace the arguments that have led to the present research rationale, building on [1].

Evolution and self-organisation Because evolution is a weak and slow process that, in order to avoid instabilities (death of the phenotype), can only make extremely small modifications to a given genotype, the ecosystem itself must already be highly performant, in the sense that its ‘components’ must already be quite compatible with one another and must already be close to satisfying a given fitness requirement. This situation results from the ultimate undirectedness of evolution, which has bootstrapped the ecosystem very slowly and through a massively parallel process. The result is a system in which hundreds if not thousands of requirements are satisfied simultaneously and compatibly with one another. Such a system depends crucially on dynamical interactions between system components and is present at multiple scales, from entire ecosystems to the cell cytoplasm.

Our objective, therefore, is to find a balance between evolutionary computing and what we are calling interaction computing (or gene expression computing). We seek an integration of the two approaches that is analogous to what DNA has been able to achieve: the same molecule is a carrier of hereditary

traits across generations whilst also guiding the morphogenesis and metabolism of the individual organism. We feel that the problem of interaction computing must be solved first, before we can hope to achieve effective evolutionary behaviour. Fig. 3 shows how the abstract concept of Interaction Computing can be instantiated into different contexts. Gene expression computing refers to the nuts and bolts of cellular pathways and how they are able to construct order and exhibit stable and robust behaviour; so it is a model oriented towards a *local* perspective. Autopoietic Computing, on the other hand, looks at *global* properties of the cell and of autopoietic systems, and tries to map these properties to computer architectures that replicate autopoietic behaviour or its subsets (such as operational closure). Autopoietic Computing is discussed in [9]. Finally, Symbiotic Computing is more specifically focused on the ecosystemic properties of interdependence and synergy, and we are interested in its possible applications to software security.

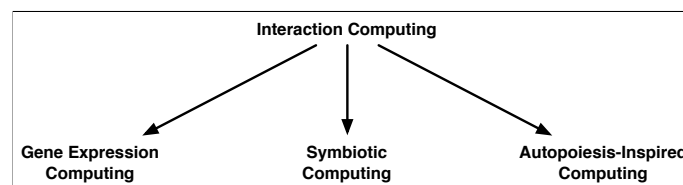


Fig. 3. Different possible models of computation derived from Interaction Computing

The prioritisation of ontogeny over phylogeny implied that an in-depth investigation of the physics and mathematics of (non memory-based) self-organisation was necessary in order to understand what features could be transferred to software. Because, in addition to the minimisation of free energy, both cell biology and ecosystems are characterised by non-linear processes, we realised that we faced a ‘double jeopardy’: not only does it seem challenging to translate non-linear behaviour into automata or algorithmic constraints, but the non-linear behaviour itself is in most cases the signature of systems that are not even integrable. In spite of the daunting stack of challenges that was taking form, we kept focusing on the fact that biological systems at all scales *are* able to cope with these challenges: they do an extremely good job at producing ordered structures and behaviour, in spite of their complexity and of the non-integrability of most mathematical models of biological phenomena. This was encouraging (if biological system can manage this, there must be a way to formalise it), even if it suggested to us that new ways of thinking about complex physical and biological phenomena were likely to be needed.

Symmetry Symmetry is a very general concept in mathematics that formalises the notion of invariance or regularity. In mathematics, a symmetry is a *transformation* that leaves some property of a mathematical object invariant. Now, since the invertible transformations of a mathematical object that leave some property of its structure invariant form a group, the mathematical study of symmetries and regularities must necessarily rely on algebra.

The above statement should be taken as a *necessary* rather than as a *sufficient* condition. In other words, a technical system that interfaces at some level with human users and that is meant to support socio-economic processes must be open to new information and must allow for the emergence of new structures and patterns. Even if such a requirement were not enforced or relevant (i.e. if all we were trying to do was to develop an artificial life environment), the wish eventually to replicate and support evolutionary behaviour implies that the emergence of new forms must be supported. Our current understanding of algebra is not necessarily sufficient to develop the best mathematical framework for the formalisation of emergent behaviour and open-ended evolution. By the same token, however, the system must also be stable and reliable, since it is meant also to uphold robust (self-healing!) engineering applications and non-functional requirements. It must behave similarly in similar contexts; hence, it must embody a fair amount of regularity and predictable behaviour. This is what mathematics, and algebra in particular, formalises. Again, we wish to emulate the delicate balance between order/reliability and unpredictabil-

ity/openness that biology has been able to fine-tune and leverage to produce stable but ever-changing life-forms.

Lie groups Since the interactions that appear to be at the basis of ecosystem or metabolic dynamics are based on strong coupling and feedbacks, our assumption and expectation is that any model of these phenomena will need to be able to address their non-linearity. In [10,5,6] we therefore began a discussion of the method of Lie groups for the solution of differential equations, since it is the most general method that applies equally well to linear and non-linear systems. At that time we were aware that a method developed for continuous systems would be difficult to apply to discrete automata, but we were also aware of the fact that generalisations of Lie groups have been applied to discrete dynamical systems.⁴ Since the Lie groups approach has not yielded generally usable results yet we will not discuss it in detail in this chapter.

Functional completeness Digital computers today are able to perform any computation because Boolean algebra is functionally complete. This means that any n -ary function can be represented by a corresponding propositional logic expression (or ‘polynomial’) that is implementable as logic gates. It has been known for many years that one can use more general algebraic structures to achieve equally functionally complete computational models. Rhodes and Mauer proved that finite simple non-abelian groups (SNAGs) are functionally complete [12]. Horvath investigated how short the realising polynomials can be [13]. Kaiser observed that a semigroup cannot have the functionally complete property unless it is a finite simple non-abelian group [14].⁵ Because, even though they are somewhat special, there are infinitely many such groups, this means that we could build a ‘more complex’ computer science using more complicated fundamental structures. Although this perspective is very promising, it is still too advanced relative to our current level of understanding of how biology ‘computes’, so we will leave it for future work.

Abstraction level Fig. 4 provides a map that may help in following the discussion in the next few sections.

Cell metabolism relies on ultimately undirected bottom-up and random/stochastic processes that can only ‘execute’ through the spontaneous interaction of the various components. The interactions are driven by a combination of electrostatic forces (usually conceptualised as minimising the potential energy of interaction) and most probable outcomes (maximisation of entropy), which can be modelled together as the minimisation of free energy [15]. In spite of this fundamental randomness, however, a healthy cell behaves in an organised and finely balanced way that is more evocative of a deterministic, even if very complex, machine than of random chaos. The cell in fact has a definite physical structure and executes well-defined ‘algorithms’ in the form of cellular processes (several hundred per cell type) such as metabolic or regulatory biochemical pathways. This suggests a description and modelling of cell behaviour at a level of abstraction that is higher than the molecular, and through mechanisms or constraints that are complementary to stochastic processes.

In particular, our perspective views the stochastic nature of cell biochemistry mainly⁶ as a mechanism of dimensional reduction that does not necessarily need to be emulated in any detail. For example, a gene expresses hundreds of mRNA molecules which, in turn, engage hundreds of ribosomes for no other reason than to maximise the probability that a particular, *single* genetic instruction will be carried out, such as the synthesis of a particular enzyme. As a consequence of this dimensional reduction (hundreds to 1), a higher level of abstraction than that at which stochastic molecular processes operate is justified in the modelling approach – in particular, a formalisation that retains, and builds on, the discrete properties of cell biology. However, even the resulting lower-dimensional system can’t plausibly be imagined

⁴ See Maeda [11] and Peter Hydon’s recent work at <http://personal.maths.surrey.ac.uk/st/P.Hydon/sym.htm>.

⁵ Every group is also a semigroup, but not conversely of course.

⁶ As pointed out by Dr Ossi Nykänen of the Tampere University of Technology, the statistical nature of cell metabolism is also important in itself, of course. For example, it carries a built-in robustness, i.e. if something is wrong with one of the proteins being generated, the metabolic cycle as a whole can proceed unhindered.

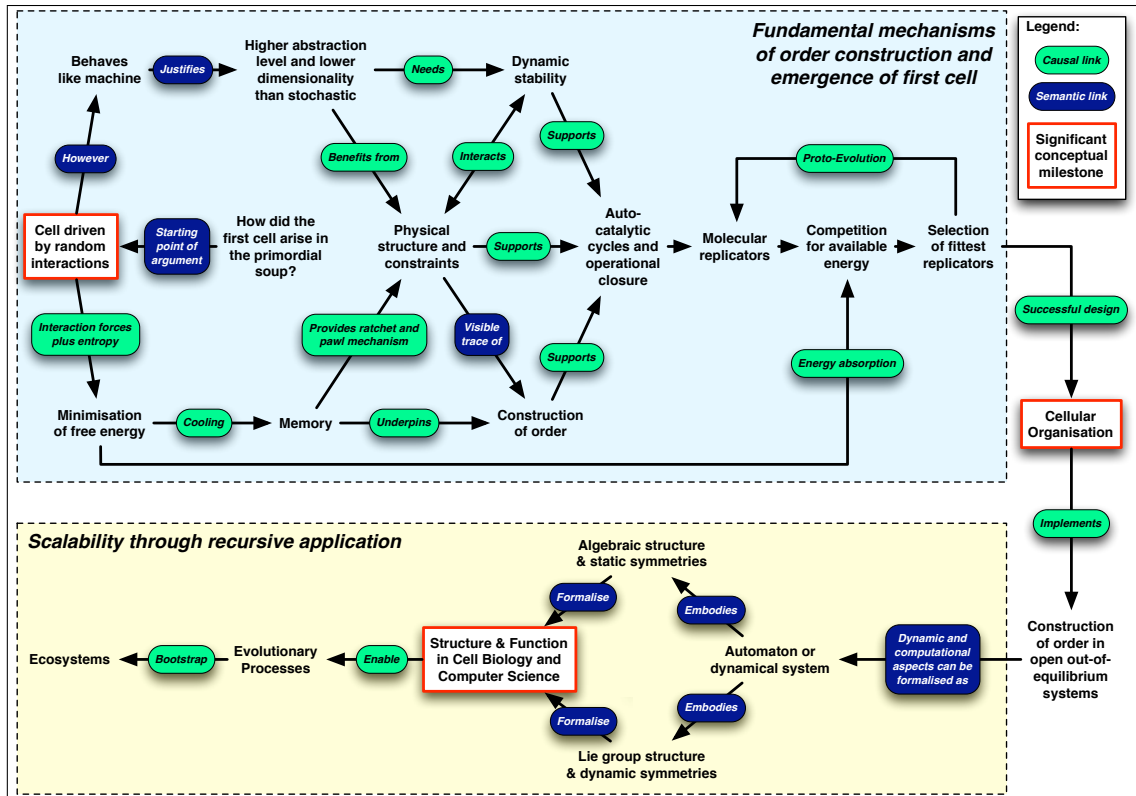


Fig. 4. Causal-semantic workflow summarising a part of the research rationale

to perform the complexity of a cell’s functions driven simply by a uniform distribution of interaction probability between its (now fewer) components. Additional structure and constraints must be at play, for example as provided by molecular selectivity. Whereas the evolutionary processes that have led to molecular selectivity and the underlying physical processes that ‘fold and hold’ the relevant proteins and molecules together are fascinating phenomena that can be recognised as the ‘efficient’ and the ‘material’ causes⁷ of order construction in biology, respectively, this does not entail that it is necessary to reproduce these mechanisms to arrive at self-organising formal systems. We think it is sufficient to recognise the *effects* of these phenomena as embodying the essence of the cell’s discrete behaviour as a kind of computation, whose ordered properties can be formalised through algebra. In other words, we believe it is sufficient to account for the ‘formal cause’ of biological behaviour in order to reproduce it in computer science. This is the motivation for our work in the development of an algebraic theory of interaction computing. A formal foundation for this theory has been provided by Rhodes’s work [16].

The fact that the cell is not a well-mixed solution tells us, as is well-known, that it must not be in thermodynamic equilibrium. Prigogine’s work [17] is deeply significant because it showed that ordered structures form in open systems under conditions of disequilibrium – maintained as such by a constant energy flow. Thus, although the phenomena he studied (e.g. the toroidal vortices of Rayleigh-Benard convection) are much simpler than what happens inside a cell, his insights give us a relatively concrete example of what a ‘dynamical structure’ might look like. The dynamic stability of cellular processes then constitutes a generalisation of Prigogine’s ordered structures. Therefore, treating cellular processes as automata, or discrete low-dimensional dynamical systems, appears to be the most appropriate level of abstraction and entry point to understand biological construction of order in a way that is relatively easy to transfer to computer science.

Structure and function in biology and computer science To make progress in this direction, we take as a starting hypothesis that the dynamically stable operation of the cell is critically dependent on two

⁷ In the Aristotelean sense.

additional forms of structure that are more abstract than physical structure and that can be formalised mathematically as follows (see Fig. 5):

- Time-independent algebraic structure of the automata modelling the cellular pathways. Algebraic structure gives rise to what we are calling static symmetries.
- Time-dependent Lie group structure of the dynamical systems modelling the same cellular pathways. This form of structure is formalised through a mixture of algebra and geometry and gives rise to what we are calling dynamic symmetries.

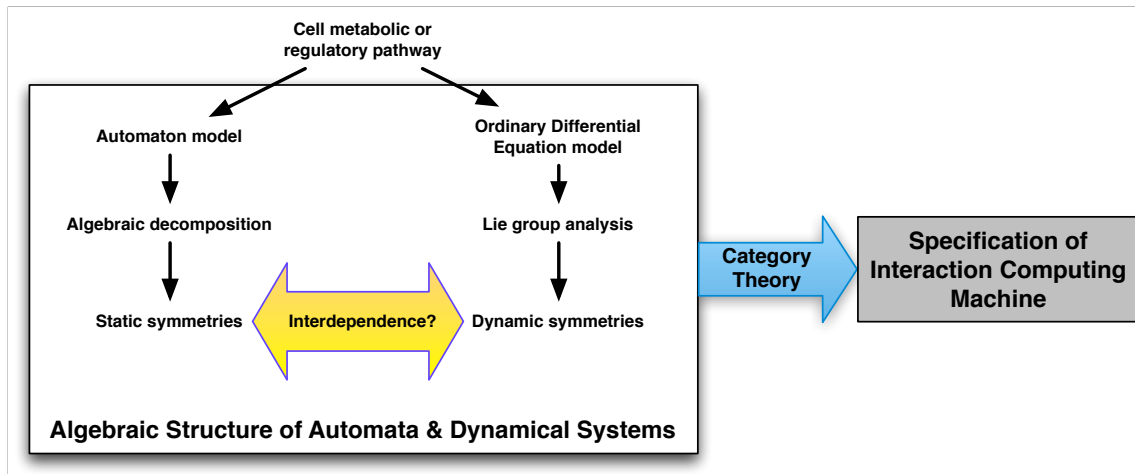


Fig. 5. Mathematical analysis workflow to uncover biological symmetries

The relevance of the relationship between structure and function to all types of engineering and applied thinking motivates us to investigate how these two kinds of mathematical structure are related. The benefit of such a relationship would be the ability to specify desired behavioural properties and derive the corresponding structural properties

Behaviour-Based Specification It appears obvious that several parts of interaction computing systems could be described by existing formal specification frameworks or formal system, such as VDM [18], Z notation [19], CCS [20], π -calculus [21], CSP [22], LOTOS [23], ACP_{τ} [24], etc. While there are languages which are very similar to our approach, and Aspect-Oriented Programming (AOP) is certainly one of them, the reason for developing a new language is fundamentally different. Interaction computing is highly different from existing systems in terms of its concurrency, its interdependability, its realisation of functionality, its non-deterministic and probabilistic computation, and its modularity. Modifications of some specification languages may support all these properties. This has been shown in the past, for example, for Z. Step-by-step the original language was extended with new features, such as non-determinism or the full support of temporal logic. This valuable engineering process extends a language such that it fits a certain need. However, this requires that the actual problem the language describes be similar.

Instead of trying to describe interaction computing using an existing language, adapting it to our needs, we take the opposite approach and start with analysing the problem first, i.e. its dynamical and structural properties. In the course of our research we will learn about this structure and identify basic functional components inspired by biology. This will also determine the primitives of the language. On top of that, our language will be based on behaviour the system to be described should exhibit. Here, the details of the internal structure of the components realising this behaviour is not essential. They are hidden from the specification as they are far too complex. This is in strong contrast with existing formal specification methods which try to describe the actual functionality but not the behaviour. Here we define functionality as the actual functions (in the sense of ‘methods’) which have to be executed to implement a certain behaviour.

Thus, the functionality of an interaction machine describes in detail the internal states and transitions the machine has to go through in order to achieve its desired behaviour, i.e. the specification would follow a white box characteristic approach. In contrast, the behaviour describes the observable or expected effects of a black box. Thus, behaviour strongly abstracts from the internal structure and gives a wider flexibility to its implementation. This takes the established high-level programming and specification languages one step further. While they already abstract from the hardware level and use higher-order programming language constructs, the biologically-inspired interaction computing specification language even abstracts from functionality and lifts programming and specification to the behavioural level. In Section 5 we study how the two concepts of machine structure and its behaviour are strongly linked in categorical terms. In particular, we show how a category of behaviour is directly linked to a category of machines realising this behaviour.

Additionally, to be able to transform an existing specification into an executable form, the specification language requires some operational semantics which allows us to translate a behaviour specification into interaction machines and their execution steps. Similar to functional or logical specification languages, the realisation of such an approach in an executable instance includes several implicit steps which are not explicitly stated in a machine specification. In interaction computing this process is even more complex because even simple operations are realised by multiple interactions between multiple machines. Adapting the operational semantics of an existing language becomes infeasible. Thus, we follow the general design process which tries to develop a language which actually fits best our needs.

Finally, we do not refuse the use of existing formal systems. In fact, our work already uses mechanisms [25] which allow us to transform one logic into a comparable one, to recognise the well-established correspondence between coalgebras and temporal logics (see also [4]), or which compare their internal structures. If we find that our systems possess properties which are describable by existing formal systems, we will opt for them, of course.

Thus, we are working towards the development of an ‘environment specification’ language, which can be seen as a higher-abstraction software engineering specification language addressing both the structure and content of bio-inspired digital systems. Fig. 6 shows at a high level how category theory can enable a mapping from algebraic and coalgebraic structures to algebraic and coalgebraic logic, as an initial step in this direction. This work is in progress and has been reported in [1], [4], and [26]. In this chapter we elaborate concepts which map algebraic structure corresponding to automata into categories of behaviour.

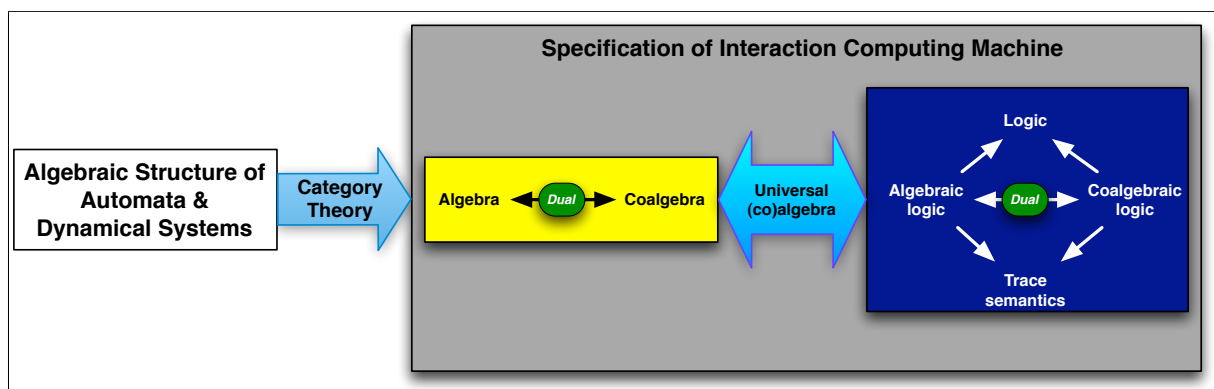


Fig. 6. Mapping of algebraic structures to logic structures through category theory

Organisation in Biology and Computer Science The reliance on category theory is further motivated by Rosen [27,28] who, following Rashevsky’s ideas [29], first applied category theory to cell biology to develop a theory of “relational biology” as an alternative to the reductionist analytical methods still prevalent to this day. His main result was to prove that the cell metabolism repair function performed by the DNA is invertible into a DNA repair function performed by the cell metabolism. Hence the cell is

‘self-sufficient’ in terms of information, it contains all the information it needs to repair all of its parts. Of course we already knew that the cell is able to repair its DNA, but for our purposes it is very good to know that the same mathematical theory that can map automata to logic and dynamical systems is also able to capture important properties of the cell. Rosen’s result has more recently been interpreted [30] as the mathematical analogue of Maturana and Varela’s “operational closure” (or organisational closure) within the theory of autopoiesis [31]. In spite of the fact that Rosen’s subsequent generalisation of this proof into a much more ambitious ‘theory of Life’ [32] has recently been criticised and has been the subject of a lively debate ([33,34,35,36,37]), Rosen should be credited with a simple but insightful observation:

... systems of the utmost structural diversity, with scarcely a molecule in common, are nevertheless recognisable as cells. This indicates that the essential features of cellular organisation can be manifested by a profusion of systems of quite different structure. [38]

In other words, all cells, regardless of their structure, share a similar organisation. However, depending on their function, cells can have very different structure. This is why we are proposing that **Structure**, **Function**, and **Organisation** are equally fundamental concepts in biology.

In computer science, on the other hand, things are a bit different. In analogue computer systems the computation to be performed (Function) was strictly dependent on the electronic components utilised and their wiring (Structure). Digital computers, by contrast, were developed as “general-purpose machines” through extensive use of abstraction/layering. In contrast to biology and analogue computers, there is very little interdependence between Structure and Function in digital computers – by design! However, Organisation does map well from biology to computer science, where it is called Architecture.⁸ An interesting example of the applicability of these concepts is provided by the “conscientious software” of Gabriel and Goldman [39], who identify software that performs some useful external function as ‘allopoietic’, in symbiotic coexistence with software that keeps the system alive as ‘autopoietic’. Thus, a concept that is similar to operational closure and that is a current focus of our research is to wire different allopoietic components together in order to form an autopoietic whole.

The complexity of the problem and of the theory that is emerging is making it difficult to keep the various analogies, metaphors, and models straight, partly because the concepts apply at very different scales and levels of abstraction. Table 1 provides a possible mapping between how these three fundamental concepts apply in biology, mathematics and computer science.

	Biology	Mathematics	Computer Science
Structure	Shape of nerve cell	Group structure of cellular pathways	Sequential/parallel/concurrent
Function	Nerve signal conduction	Metabolic pathway	Algorithm, behaviour
Organisation	Operational closure	Group closure property	Autopoietic architecture

Table 1. Examples of how the fundamental properties of biology might map to other domains

Gene expression computing, or interaction computing In reference to Fig. 4, proto-evolutionary mechanisms in the primordial soup bootstrapped resilient organisational forms such as hypercycles [40] and autocatalytic cycles [41] from random physical interactions. After the membrane emerged as a structure that could delimit an ‘inside’ from an ‘outside’, these so-called molecular replicators eventually led to the emergence of the cell with its autopoietic properties (organisationally closed, recursively self-generating). As we argued above, cellular pathways today are still driven by the same interaction and entropic physical processes. Thus, if we wish to emulate, in software, principles from biology that can

⁸ The suggestion that in the context of bio-computing biological organisation maps to software architecture is due to Prof T V Prabhakar of the Indian Institute of Technology, Kanpur.

rightfully claim ‘fundamental’ status, in its most general form context-sensitivity must work both ways, which argues for a reciprocal and pervasive interaction model.

Our work is inspired by the observation that the computation performed by a biological ecosystem can be conceptualised as a theoretical limit characterised by the number of peers in a distributed P2P architecture approaching infinity, with the amount of traditional computation performed by each approaching zero. This analogy can also be extended to the ‘computation’ performed by the cell’s cytoplasm. More precisely, the computation performed by biological systems always involves at least two entities, each of which is performing a different, and often independent, algorithm which can only be advanced to its next state by the interaction itself. This is the kernel of the concept of interaction computing or gene expression computing. We wish to explore the implications of such a ‘vanishing CPU’ scenario because by providing a mathematical foundation to building nested and recursively interacting structures we believe that it underpins a model of emergent computation that will lead to new insights in biology and computer science, in equal measure.

This hopefully explains why we are trying to develop an emergent model of computation by mapping the regulatory and metabolic biochemical pathways of the cell to interacting automata. Such a model of computation will both require and enable a shift from a reliance on human design as the only source of order in software towards a greater reliance on information and structures built into the environment. In fact, the complexity of the cell’s interior suggests that in the cell ‘interaction’ can acquire significantly greater semantics than, for example, perfect collisions between point particles in an ideal gas. We then notice that the cell is itself surrounded by other cells with which it communicates, and all are embedded in a complex mixture of tissues and fluids that form organs. Organs, in turn, cooperate in the functioning of individuals, which interact to form biological ecosystems. Thus, interactions happen at all scales within the nested and recursively organised hierarchical structure of all biological systems.

3 Recent Insights

In this section we elaborate in more depth some of the concepts and ideas summarised in the previous.

3.1 Oscillations, cycles and groups

The evidence so far suggests that a stable pathway is not stable of its own accord, but is kept within the analogue of a stable ‘potential well’ by the pathways it is biochemically coupled to, which therefore can be seen as constraints. As a consequence, it appears that in order to achieve the Interaction Computing vision we will have to understand how multiple threads, that are performing different algorithms, need to be coupled so that they can aid or constrain each other, as the case may be.

The predominant mathematical model used to analyse the interdependence of biochemical pathways is a set of coupled, and generally non-linear, ordinary differential equations (ODEs) derived from the chemical reaction equations. The set of dependent variables in such a set of ODEs is made up of the concentrations of compounds participating in the chemical reactions. Starting from these same chemical reaction equations, the system dynamics can be discretised as a Petri net, from which a finite-state automaton can be derived. It is possible that the ‘dynamic stability’ exhibited by cellular pathways is somehow related to the attractors in Kauffman’s Random Boolean Networks, which he originally introduced in the 1960s to model gene regulatory functions [41], but we have not explored this path yet.

For the present we have focused on the p53-mdm2 regulatory pathway and on the analysis of its discrete and Lie symmetries. The details of the Lie group analysis of this system are discussed in [6]. P53 is a protein that participates in most regulatory pathways of the cell, and mdm2 is another protein that regulates the concentration level of p53. Depending on the p53 level, the cell can suspend its cycle to allow its DNA to be repaired, it can reach the state of reproductive senescence (completing its cycle but not reproducing), or it can immediately destroy itself. The malfunction of the p53-mdm2 system is believed to be related to approximately 50% of all cancers. It is therefore an interesting case to study in the development of a theory of autonomic computing, and for this reason we have been analysing it.

As discussed in [9], the p53-mdm2 regulatory system is characterised by oscillatory and non-oscillatory regimes. A simplification of the p53-mdm2 system we have been working with assumes that in the absence of DNA damage the response of the system to non-equilibrium starting values of p53 and mdm2 is to create a peak of p53-mdm2 complex until enough p53 is destroyed and its level is brought back to equilibrium, without oscillations. On the other hand, if phosphorylated p53 (p53*) is present because of DNA damage, then the system responds with a damped oscillation in its p53 level, until the damage is fixed. This behaviour is confirmed by the mathematical analysis based on Lie groups that is discussed in [6]. There are two ways in which Lie groups can help us with this problem: (1) by helping us solve the system of equations; or (2) providing additional information on the symmetry structure of the problem. So far we have only benefited from the first one, in particular by deriving a single first-order Riccati equation from the original system of four ODEs.

Although in the long term we believe that greater insight will be gained by understanding the relationship of any Lie symmetries that might be present to the observed dynamical behaviour, in the meantime we have developed an intuitive understanding of the relationship between the algorithmic symmetries of several biological pathways and the algebraic structure of the automata derived from them. In particular, cyclic or periodic behaviour in the physical/biological system appears to correspond to the presence of permutation groups embedded in the semigroups associated with the automata derived from these systems. Therefore, we expect to see more permutation groups in the model that corresponds to the oscillatory regime of the p53-mdm2 system than in the model that corresponds to the ‘healthy’ regime. Fig. 7 shows the automaton corresponding to a 2-level Petri net derived from the biochemical equations of the p53-mdm2 system [42]. More discussion on the relationship between the oscillatory nature of biochemical systems and permutation groups is provided in [43].

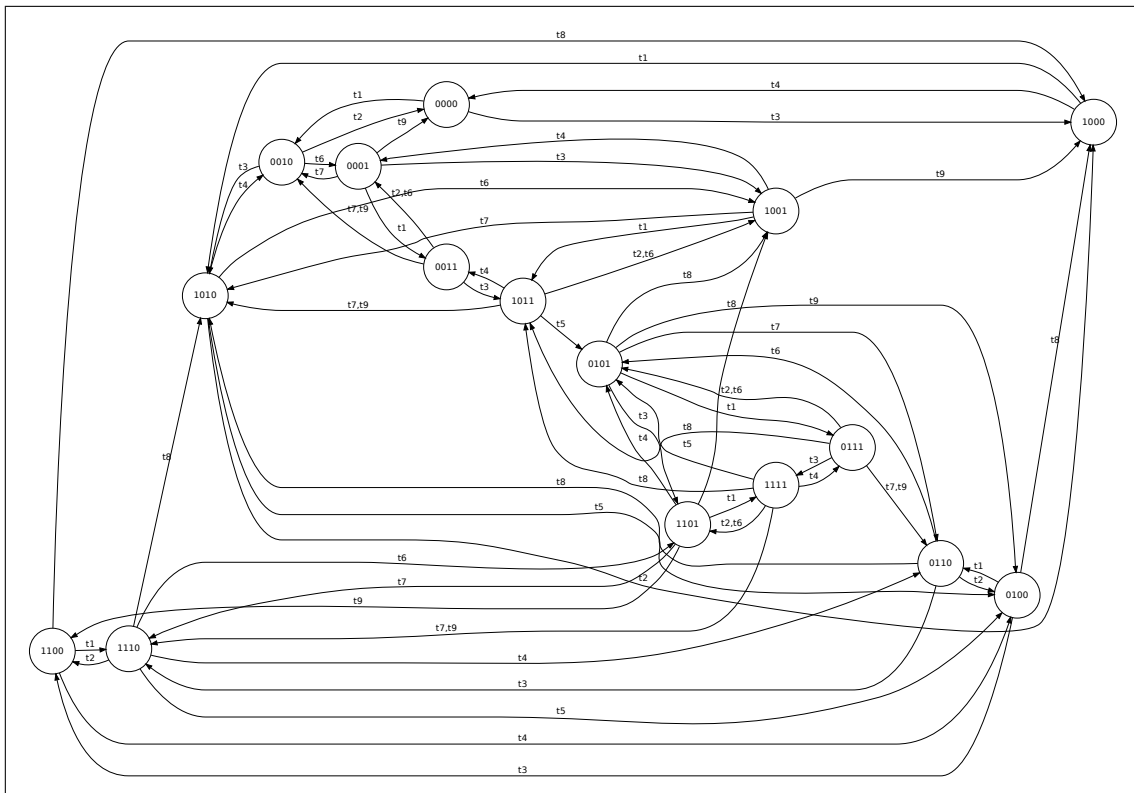


Fig. 7. 16-state automaton derived from 2-level Petri net of the p53 system

3.2 Scale dependence of interactions

The mechanism of interdependence between the different systems that comprise a biological organism and that ensures its smooth operation is far from obvious or well-understood, but we can begin to postulate a possible scenario. For example, the manner in which it might apply to symbiotic security presents at the same time a challenge and an opportunity. The challenge arises from the fact that it is undesirable to expose to an end-user the possible security options that may be relevant to a given application. Even when the ‘end-user’ is him/herself a developer of services and applications, ideally the security features should be automatically guaranteed, rather than having to be specified in detail – something a security expert is best positioned to do. But it is far from obvious how such security guarantees or ‘qualities’ can be achieved without an explicit specification. The opportunity comes from the observation that in biology systems and sub-systems are not controlled ‘from above’; therefore, the most likely source of control of a given system must come from the other systems it is coupled to. We are beginning to understand how biology achieves this, as we discuss below, and thus we may be able to begin to address the challenge of how to achieve transparent security guarantees.

The longer-term view of this approach is ontogenetic, which requires a two-step design methodology. The first step requires the participation of security experts and is concerned with developing an abstract specification of desired interdependence between the two classes of services (applications + security services). In this view, such qualities are encoded in the more concrete specification of an *environment* that can support the ‘breeding’ of the two instances through a process analogous to morphogenesis. The second step, which ultimately becomes the one most often utilised, requires the application developer to specify their application using a language that can be compiled for the cytoplasm-like environment which embodies the security qualities, in order to achieve an instantiation process able to integrate seamlessly the security service into the application.⁹ A further step, which is not the focus of our current work, would be to wrap such a development environment in an evolutionary framework able to adjust in small steps the first specification (of the ‘digital cytoplasm’) based on run-time feedback and on changes occurring in the wider web environment.

Such a developmental and ‘emergent’ perspective, however, is too advanced for the current state of the art. We therefore opted for the less ambitious but still daunting task of **characterising the fundamental structure and behaviour** of an already ‘grown’ system that is ready to perform the two interdependent functions (security + other function) from the moment it is instantiated. This characterisation is to be integrated in the familiar engineering methodology of deriving the required structure from the formal specification of the desired behaviour. Such an approach, even if simpler, would still be able to address the *ex post* patching problem.

Formalisation perspective The research towards the above high-level objective was very tentative at first, but slowly converged around three sets of ideas that emerged spontaneously, and that naturally came together into a promising framework:

- ‘Characterisation of structure and behaviour’ specifically implied the desire to achieve predictable, reliable, and self-healing software constructs and dynamics. The constancy or invariance in structure and/or behaviour that these terms imply points to the need for an algebraic perspective. This was confirmed by the extensive work in algebraic automata theory that has recently received a surge of attention [44,45,16], perhaps because this rather mathematically-oriented work is only now beginning to approach more applied examples in computer science.
- As we mentioned above, whereas biological *diversity* can be seen as a consequence of the breaking of some of the symmetries underpinning physical systems, biological *order construction* is fundamentally reliant on preserving the same (or different) symmetries. Therefore an evolutionary framework for bio-computing must find a similar balance between variation and constancy. Our approach has been to prioritise the latter to achieve ordered structures and behaviour before attempting to perturb such order to achieve variation (through e.g. an evolutionary process). The relevance of the same

⁹ We should use words like ‘seamlessly’ with caution. This is explained in more detail in Section 3.2.

mathematical concept of symmetry to software and to biology meant that the same algebraic approach could in principle be used to analyse biological behaviour and then to translate the findings into computer science formalisms. This hunch was at least partially confirmed by an extensive body of research that originated over 40 years ago [46,47,48,49,50] and that we were able to begin collaborating with during the course of this work.

- In the middle of this work we became aware of the important and increasingly pervasive role of category theory [51,52] in connecting very disparate fields (such as e.g. cell biology, logic, algebra) and realised that it could provide a unifying formalism and framework whereby algebraic properties of automata structure could be transformed into algebraic properties of automata behaviour [53,54,55,56,57] which, in turn, could then be further transformed into the skeleton of a specification language [58,59].

In other words, category theory helped us see how these complementary theoretical areas of investigation could be integrated into a single and plausible theoretical framework.

Towards an epistemology of non-linear science While this formal landscape was taking shape, we were continuing to toy with the deeper and older idea that, with the exception of high-level cognitive processes, construction of order and ordered behaviour in biology are not *directed* or *planned*. Both arise spontaneously out of random interactions between the various components and constituents of biological systems at many different spatio-temporal scales. This observation implies the need for a shift in thinking. For example, referring to the immune, nervous, and endocrine systems mentioned above, ‘interdependence’ means more than the need for each system to be compatible with the others, or to get a ‘green light’ from the others in order to advance its processes.

It is difficult to generalise or to say anything absolute about biological systems since, having evolved randomly and opportunistically, in a ‘value-free’ manner, more often than not they embody a *mixture* of what from an engineering perspective could be called ‘solutions’ and that are generally very challenging to unravel. By contrast, when designing a complex system the engineering approach attempts first to classify the system into separate components or sub-systems according to their structure and/or function. This is essentially an *analytical* step that has dominated the traditional scientific disciplines since the Greek philosophers. In the next, *synthetic* step, the engineering approach generally tries to minimise the interdependencies between the components or subsystems in order to simplify the control of the overall system. As a consequence, when traditional analytical and synthetic methods are applied to the analysis of biological systems or to the construction of models thereof, they can be criticised for being ‘reductionist’, i.e. of losing track of the forest by focusing too much on the trees. This is precisely the criticism that the 2nd-Order Cybernetics, General Systems Theory, and Autopoiesis movements brought to traditional science [29,32,60,61,31,62,63].

While we are very sympathetic to the ideals of the more holistic scientific theories, we can’t help lamenting the dearth of formal, explanatory, and quantitative elements that characterises most of these theoretical attempts. We therefore propose a compromise. We believe that the traditional *analytical* step is very valuable for improving understanding. Taking this step does not take anything away from the holistic properties of the system *as long as we refrain from enforcing the traditional engineering synthesis based on the separation of components’ structures and functions and the centralisation of control*. Therefore, this is where we must come up with an alternative, which we are very loosely stereotyping as *non-linear* since it is inspired by the complex couplings and interactions we observe between the components of biological systems seen as dynamical systems.

The approach we propose here can only be a partial answer because the non-linear part of the story is still unsolved. Our intention is to complement, or possibly even replace, the physics perspective on non-linear systems as low-dimensional *continuous* phenomena governed by systems of coupled ordinary differential equations (ODEs) with a *discrete* mathematics perspective of interacting automata whose algebraic structure is derived from biochemical pathways. We know that the available analytical techniques used to study systems of ODEs are of limited power relative to the complexity of most biochemical systems (e.g. the Lie groups methods discussed in [5,6]). Interestingly, we are even farther behind in characterising the dynamical properties of computation in anything resembling an analogous manner.

However, we still feel that our perspective is worth investing in because by attempting to relate the interaction characteristics between automata to the global system behaviour it is attempting to perform the synthetic step discussed above in a manner that is ultimately analogous to mathematical integration.

A constructive methodology (language, i.e. programming) rooted in the discrete mathematics perspective of computer science (i.e. algebra) might succeed where the very sophisticated non-linear dynamical systems theory and Lie group methods are struggling. In essence this implies the need to continue pushing towards a theoretical synthesis between the epistemologies of language and of mathematics, beyond what computer science has been able to achieve so far. As such, it is the only way we can see in which an explanatory, holistic, and quantitative theory of complex discrete/computational systems can be developed.

Dynamical perspective With the above provisos, as part of the analytical step we propose an idealised system model where the operation of the automata components can be separated and rationalised into 3 different functional categories:

- The *driving force* of the computation is provided by the minimisation of free energy. This is easily abstracted by states that transition spontaneously when appropriate guards have been satisfied, i.e. without needing a clock input [64].
- The state transitions of a particular automaton can also be *triggered* by the automata it is coupled to.
- The triggers between automata are not ‘agnostic’ but mimic the specificity of enzymes and *carry well-defined semantics*.

The identification and assignment of these functional roles to appropriate components or subsystems aim to provide something similar to a ‘scaffolding’; or, better, an inner structure and *vis vitalis*¹⁰ as an alternative to our default conception of order construction through an anthropocentric, centralised, and top-down supervisor.

For example, when thinking about coupled systems in the biological context, we are not generally concerned with the mechanism by which each of them is driven to execute its functions. If pressed, we might imagine an involuntary process, perhaps housed in the cerebellum, that acts as ‘supervisor’ and sends nervous or hormonal signals to keep everything running smoothly. When translated into the abstraction of coupled automata, it is not hard to envisage some global clock that causes each system to advance in the execution of its processes. This, however, is an anthropocentric interpretation, which is influenced by the unspoken and implicit assumption that at some level a cognitive or control system-like process will be needed. In fact, biological systems in the great majority of cases rely on a ‘distributed intelligence’ in the sense of [2]. At ‘simpler’ levels of description such as the interaction of sub-cellular systems, the same anthropocentric bias pushes us to think of the DNA as the ‘controller’ of all the cellular processes.

It is worth noting that our proposal partly departs from Maturana and Varela’s “structural coupling”, which is defined in two steps:

Structural Determination. A process of change of an organism that, at any point in time, is determined by the organism’s previous structure but is triggered by the environment. The same holds for the environment: the organism is a source of perturbations and not of instructions.

Structural Coupling. A form of interdependence between two actors or entities that satisfies the criterion of structural determination mutually and symmetrically. [62]

Thus, structural coupling is equivalent to the first two bullets of our proposed model. In addition, it places a great emphasis on the individual past histories of the components: it is the past history that determines the future evolution, not the ‘other’ component as we are saying here. By proposing a different mechanism we are not overlooking the importance of path dependence, which is of central importance in all dissipative systems, i.e. all real systems. Rather, we are shifting the emphasis from memory-dependent processes to non-linear interaction processes. It is no coincidence that Darwinian evolution and neural networks construct order through memory-based processes. Although we have not yet developed a full argument and theoretical justification, we think that scale plays an important role in determining which of these two kinds of order construction processes dominates in biology: at smaller scales where e.g. globular enzymes spin on their axis at 50 thousand times a second and substrates zip by at several metres/second [65] the world is more ‘mechanical’, and non-linear energy or information coupling

¹⁰ Life force.

mechanisms dominate; at larger scales, which depend on the construction of durable structures, past history and memory mechanisms become more important. Our approach, with the introduction of the third bullet above, wishes to take the non-linear dynamics as far as it will go, before starting to rely on memory (which is ultimately founded on phase transitions and therefore on a statistical description of the system [15]). Indeed, the richest non-linear behaviour uncovered by Chaos theory is exhibited by *dissipative* non-linear systems, thus a **multi-scale approach** that is able to account for all these effects will ultimately be unavoidable, as summarised schematically in Fig. 8.

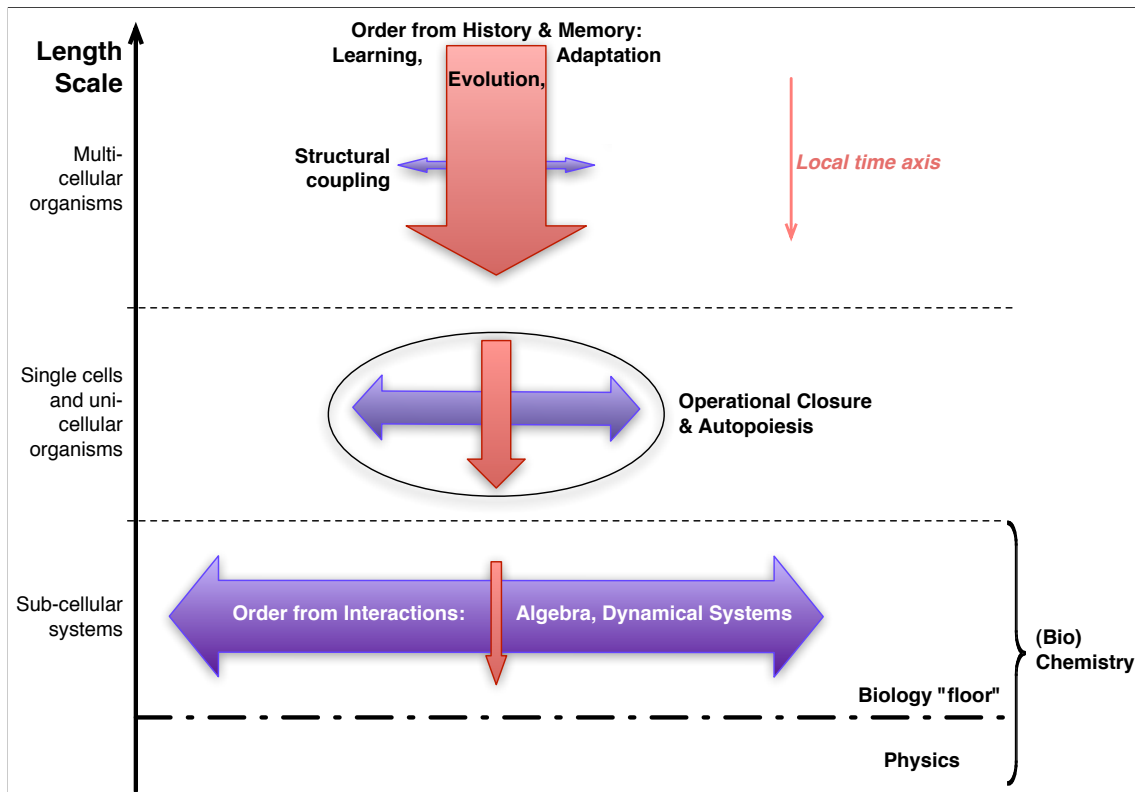


Fig. 8. Schematic showing the different relative importance of order construction mechanisms in biology, at different scales of description

The tighter interpretation of interdependence that we believe dominates at smaller length scales in biology and that enables bottom-up, unsupervised order construction means that each system *cannot operate* if the other systems are not also operating. An even more extreme view is that each system is being *driven* by the others. This seems to be captured well by the timed automata discussed in [6], where interdependence between automata is modelled through system variables (“guards”) and clock constraints that need to be satisfied before the automaton can continue out of its current state. Whereas the clock constraints appear somewhat artificial and imposed ‘from above’, the guards could mimic the interaction between different biochemical pathways in the sense that Pathway B, for example, cannot proceed until a particular reaction product is produced by Pathway A. The main point is that the timed automata discussed in [64] advance to the next state *spontaneously* when all the guards and constraints are satisfied, they do not need a clock input.

Theoretical framework for interaction computing Thus, in looking for general principles that appear to govern self-organising behaviour in cell biology, in particular at the sub-cellular scale, the role of interactions appeared to be such a fundamental feature that it seemed indispensable to replicate it in computational systems in order to develop an ‘architecture of self-organisation’ in software along analogous principles. In particular, the addition of the algebraic automata theory perspective to the study of biochemical systems has opened the possibility to develop a formalism that can express the behaviour

of **biological systems** seen as **dynamical systems** in a manner that is consistent with **the mathematical foundations of computer science**. We are pursuing this research vision through the development of the concept of interaction computing [4,5,6,66].

Fig. 9 shows how the concepts presented so far in this section can be fitted together at a high level within the same theoretical framework.

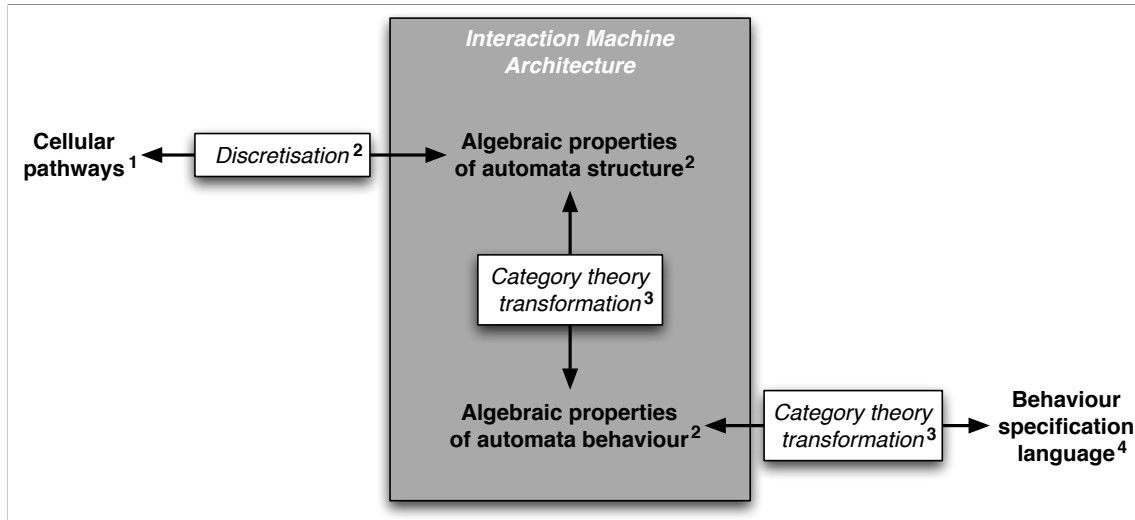


Fig. 9. Theoretical areas of relevance to interaction computing. Legend: 1 = [67]; 2 = [43]; 3 = [4,5,6,7,66]; 4 = future projects/papers.

Why algebra (again) The formal framework and run-time system discussed in [64] is very general and would appear to be expressive enough to be able to model the automata discretisation of multiple, coupled biochemical pathways (although we have not tested this claim). In other words, if we adopt a position of scepticism what we have said above about interaction computing is not essentially very different to the timed automata point of view. Hence, what does interaction computing actually add that is different and (hopefully) useful? The short answer is a shift in the source of the encoding of behaviour, from a central supervisor to the components.

We argue that where the timed automata perspective requires the knowledge ahead of time of the behaviour of every component of the system, and of their interdependencies, the interaction computing perspective uses the interdependence as a *trigger* and a *lever* rather than as a *constraint*. The difference is subtle but important. A constraint *prevents* something from happening or, if it is met, *allows* the next event. A trigger *initiates* the next event, and a lever *drives* it. In the biochemical case the latter point has a precise physical meaning as in, for example, the provision of energy-carrying ATP molecules (produced by the mitochondria in the Krebs cycle) to other pathways in the cell. The implication is that a particular action can initiate in one automaton, and as this automaton executes the automata it is coupled to could likewise undergo state transitions by virtue simply of the coupling between them.

In the absence of an omniscient supervisor, coupled automata (systems) must serve also a third function for each other, as listed in the previous sub-section: they must provide information or, in computational terms, semantics. In other words, although the operation of the cell is ultimately random and unplanned, its fine-tuned clockwork-like operation suggests an interpretation based on internal cause-effect relationships between its component parts. In other words, we are back to Cartesian rationalism – to some extent. Although the interlocking pathways probably do not *determine* each other's behaviour, we argue that entropy-maximisation processes do approximate a certain level of determinism (as we did also in [66]).¹¹

¹¹ The irony of a Cartesian process being approximated by a Gaussian one is not lost on us, but we will have to postpone a more in-depth analysis and verification of this rather surprising implication of our argument.

This means that each transitions spontaneously to its next state when ready, and each acts as a trigger and a source of semantics for the other. But then, with this construction we are simply describing, for each half of this coupled system, the action of a semigroup on a set of states, i.e. of a ‘transformation semigroup’ [43]. In other words, this argument seems to be leading us inexorably towards an algebraic conception of the problem. More precisely, the *output* alphabet of Automaton A with the operation of concatenation forms a semigroup that acts on the states of Automaton B in the normal way, i.e. through Automaton B’s *input* alphabet with the operation of concatenation; and vice versa.

Although a larger system could be envisaged that encompasses both automata and all their interactions, thereby recovering a Turing-computable system, the point of interaction computing is that each automaton may be driven additionally by other inputs that come from independent sources. Although the Turing machine argument can be applied recursively to ever-larger systems, it quickly loses relevance once we realise that the individual sources of inputs can also be different human users interacting with the same system. Although the Turing machine perspective is still applicable *ex post* [68], at this level it becomes rather an empty academic argument: what’s more interesting and pressing is how to build a dynamical system that can produce useful behaviour when subjected to independent and unpredictable, i.e. not programmed, inputs by an arbitrary number of sources – precisely as biological systems seem to be capable of doing (see [6,43] for more discussion on this point).

We may now finally be able to explain in what sense we meant the qualifier ‘seamlessly’ at the beginning of this section. Because each component (automaton, service, etc) relies on the components it is coupled to for instructions on what to do next, the algorithms executed by each automaton are semantically interdependent, like interlocking pieces of a dynamical puzzle, not simply triggering each other’s transitions like a clock input might do. In other words, because the output symbols that drive Automaton B are generated by the states visited by Automaton A, this means that the instructions that drive Automaton B are *encoded in the behaviour* of Automaton A. In the interaction computing view, the algorithm of each automaton is overloaded: the algorithm of Automaton A performs what Automaton A is supposed to do, but it encodes also what Automaton B is supposed to do. If we allow for a complete and symmetrical reciprocal interdependence we end up with a chicken-and-egg problem; so perhaps – again – a mixture and balance between previously specified behaviour and dynamically determined instructions must be reached.

In conclusion, it is possible to envisage a particular sequence of state transitions representing the behaviour of a software service not only triggering but instructing the behaviour of a complementary service that could implement the corresponding security functions for that algorithm, and vice versa. The algebraic and category-theoretical framework for how this might be done is beginning to take shape and will be pursued in future projects that build on the BIONETS outputs.

3.3 An initial model for interaction computing

Many years after proving the prime decomposition theorem for semigroups and machines, John Rhodes published a book that he had started working on in the 1960s and that has come to be known as the “Wild Book” [16]. In this book he provides a very clear definition of an alternative to a Turing machine, which we believe to be a very promising starting point for a model of interaction computing. The material in this section is taken from [6,43].

As we know, an algorithm implementable with a Turing machine is equivalent to the evaluation of a mathematical function. As Wegner and co-workers argued in a series of papers over the last 20 years ([68] and references therein), the evaluation of a mathematical function can afford to take place by following an ‘internal clock’, i.e. the Turing machine is isolated from its environment while it evaluates the algorithm. Biological systems, on the other hand, are continually interrupted by external inputs and perturbations. As an example of this class of computations Golding and Wegner used “driving home from work”, which they described as a non-Turing-computable problem. Turing himself had foreseen this possibility in his original 1936 paper as the “choice machine” [69], although he did not pursue it further.

Similarly, Rhodes defines a machine as a special mathematical function. Unlike a regular function from a set of inputs to a set of outputs, Rhodes’s function or sequential machine accepts an input at each discrete point in time and generates an output based on its state and on all the previous inputs up to that

point. The realisation of such a machine is achieved through a finite-state automaton that Rhodes calls a “circuit”, $C(f)$. He goes on to say

The reason why Turing machine programs to realise a computable f are not unique and the circuit which realises the (sequential) machine f (namely $C(f)$) is unique is not hard to fathom. In the sequential machine model we are given much more information. It is ‘on-line’ computing; we are told what is to happen at each unit of time. The Turing machine program is ‘off-line’ computing; it just has to get the correct answer – there is not time restraint, no space restraint, etc. ([16]: 58)

To develop this discussion further, it is helpful to provide a few familiar definitions.

A **finite automaton** can be defined as $\mathcal{A} = (A, Q, \lambda)$ where A is the set of *input symbols*, the *input alphabet*; Q is the set of *states*, the *state space*; and λ is the *state transition function* $\lambda : Q \times A \rightarrow Q$. Since we are talking about a finite state automaton, all the objects involved are finite.

Let A be a non-empty set. Then $A^+ = \{(a_1, \dots, a_n) : n \geq 1 \text{ and } a_j \in A\}$. A **sequential machine** is by definition a function $f : A^+ \rightarrow B$, where A is the basic input set, B is the basic output set, and $f(a_1, \dots, a_n) = b_n$ is the output at time n if a_j is the input at time j for $1 \leq j \leq n$.

The sequential machine is clearly related to the definition of the finite automaton, but there is a twist: Rhodes prefers to make a sharp distinction between a *machine*, which he equates to a *mathematical function*, and the *realisation of that machine*, which he calls a *circuit* and that is essentially an automaton:

Mathematical concept and its realisation
Machine or mathematical function	Circuit or automaton
Automata behaviour	Automata structure

Due to the need to maintain the development of a theory of interaction computing on firm mathematical grounds, we follow his approach. More specifically, the separation between a machine and its realisation matches well the distinction between the description (formalisation) of *behaviour* and the automaton *structure* necessary to achieve it. It is essential for us to maintain this distinction in light of the part of our research discussed in later sections of this chapter which applies categorical morphisms to automata behaviour in order to derive a specification language. To understand better what we might be aiming to specify, let’s develop the idea further.

We are going to use a generalisation of the sequential machine, also by Rhodes, which produces an output for each input it receives and not just in correspondence of the most recent input. We are going to call this generalisation an **interacting machine**:

Let $f : A^+ \rightarrow B$ be a sequential machine. Then an **interacting machine** $f^+ : A^+ \rightarrow B^+$ is defined by $f^+(a_1, \dots, a_n) = (f(a_1), f(a_1, a_2), \dots, f(a_1, \dots, a_n))$.

Thus, an **Interaction Machine (IM)** can be built by joining two or more **interacting machines**. Such an IM will still accept inputs from outside itself (‘the environment’) and will produce outputs for the environment. The realisation of either machine is achieved through a finite-state automaton that Rhodes calls a circuit, C , but that in the literature is more commonly called a Mealy automaton:

$C = (A, B, Q, \lambda, \delta)$ is an **automaton** with basic input A , basic output B , states Q , next-state function λ , and output function δ iff A and B are finite non-empty sets, Q is a non-empty set, $\lambda : Q \times A \rightarrow Q$, and $\delta : Q \times A \rightarrow B$.

Having established then that the problem of computation is posed in two parts, a mathematical function and its realisation, we continue to rely on Rhodes to define a few more concepts related to the latter, in order to develop a relatively concrete working terminology. The next concept we need is the realisation of an algorithm, as follows. Let $C = (A, B, Q, \lambda, \delta)$ be an automaton. Let $q \in Q$. Then $C_q : A^+ \rightarrow B$ is the *state trajectory associated with state q* and it is defined inductively by

$$C_q(a_1) = \delta(q, a_1) \tag{1}$$

$$C_q(a_1, \dots, a_n) = C_{\lambda(q, a_1)}(a_2, \dots, a_n), \quad \text{for } n \geq 2. \tag{2}$$

We say that C realises the machine $f : A^+ \rightarrow B$ iff $\exists q \in Q : C_q = f$. By a simple extension of the above definitions it is fairly easy to see that the output of an algorithm can be associated with a sequential machine when the output corresponds to the result after the last input, whereas it is associated with an interacting machine when there are as many outputs as there are inputs:

$$C_q(a_1, \dots, a_k) = b_k, \quad \text{for } k = 1, \dots, n \quad (3)$$

$$C_q^+(a_1, \dots, a_n) = (b_1, \dots, b_n). \quad (4)$$

Rhodes then introduces more formalism to define precisely a particular automaton that realises a function f as $C(f)$, and goes on to prove that $C(f)$ is the unique minimal automaton that realises f . He goes on to say

The reason why Turing machine programs to realise a computable f are not unique and the circuit which realises the (sequential) machine f (namely $C(f)$) is unique is not hard to fathom. In the sequential machine model we are given much more information. It is ‘on-line’ computing; we are told what is to happen at each unit of time. The Turing machine program is ‘off-line’ computing; it just has to get the correct answer – there is no time restraint, no space restraint, etc. ([16]: 58)

Rather than constructing dynamical behaviour through a sequential algorithm expressed in a programming language, which can be realised by a single automaton or Turing machine, we are talking about constructing dynamical behaviour through the interaction of two or more *finite*-state automata. This is because, if the possibility that Q be infinite is left open as the above definition does, “then the output function could be a badly non-computable function, with all the interesting things taking place in the output map δ , and we are back to recursive function theory” ([16]: 59). Therefore, we note the interesting conclusion that, for a tractable approach, Q must be finite and that the realisation of an Interaction Machine must be made up of interacting finite-state automata. Thus, from the mathematical or behavioural perspective, we will build the Interaction Machine by using sequential (or interacting) machines as basic units and by combining them in various ways.

The mathematical and computer science challenge, therefore, is to develop a formalism that is able to capture the non-linear character of the dynamics of an arbitrary number of coupled metabolic systems, so that when this mathematical structure is mapped to our yet-to-be-developed specification language we will be able to specify software *environments* capable of supporting self-organising behaviour through interaction computing, driven by external stimuli from users or other applications.

In the Wild Book, after completing the presentation of the algebraic theory of automata that he partly founded, Rhodes analyses the Krebs cycle in great biochemical and mathematical detail, and on the basis of this discussion proposes an abstract model of the cell, as shown in Fig. 10. This model can be regarded as a more detailed instantiation of Rosen’s M-R system [4] in a computer science context, but developed from a different mathematical viewpoint and firmly anchored in cell biochemistry and algebraic automata theory. For this reason we regard this model as the end-point of the conceptual development of the IM. From this sound mathematical basis, we feel that an automaton realisation of the IM can be attempted, with the objective of reproducing biological behaviour in computer science applications such as security and service composition. For example, the pathways shown in Fig. 10 could ultimately be related to the BIONETS Service Individuals as a design pattern.

A significant amount of mathematics is still to be done in order to derive automata structure from desired behaviour. In the meantime, having reached a greater clarity on the architecture of the IM, we are now in a position to begin the mapping of this structure to classes of possible behaviours that are compatible with it using category theory, in order to develop a behaviour specification language that can be transformed into biologically-inspired structures that realise it. This approach is discussed in the next section.

3.4 Category theory and automata

In Sections 5 and 6 we will discuss category theory by introducing the most important concepts and sketching how category theory can be used to establish the important links between languages, automata, logic, semantics, and computational models in general. Consistently with the iterative presentation style

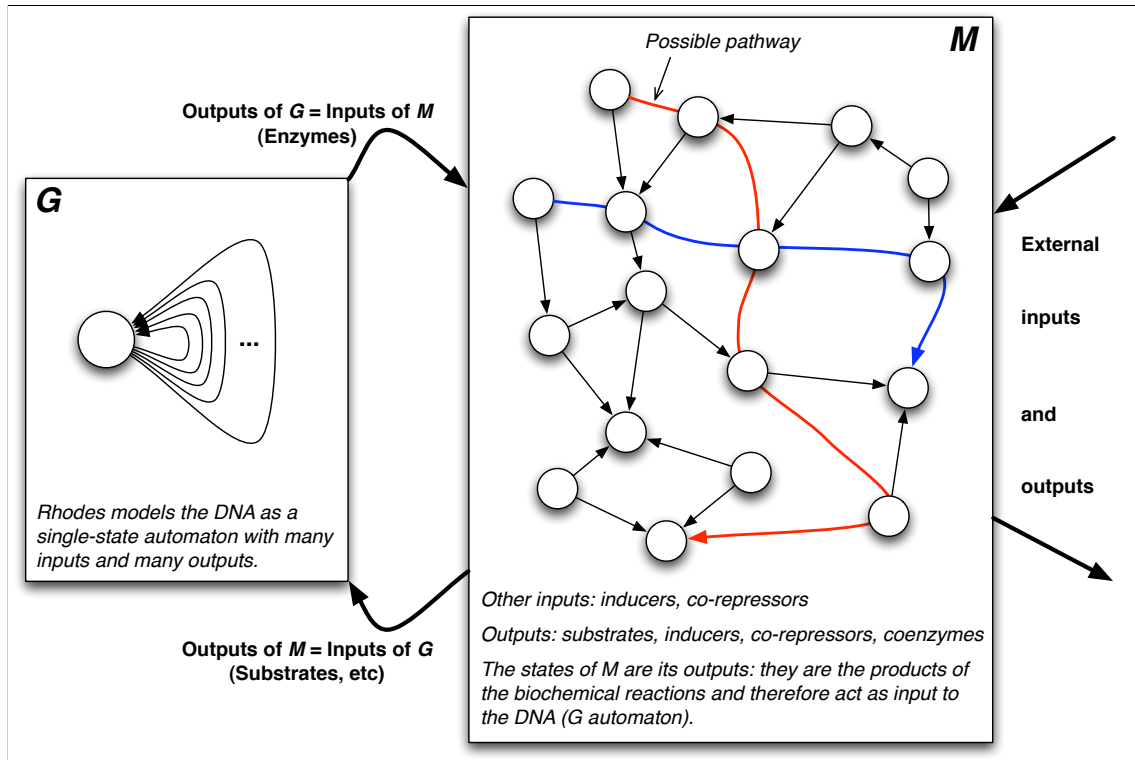


Fig. 10. Conceptual architecture of the biological realisation of the Interaction Machine, based on Rhodes ([16]: 198).

we explained in the Introduction, in this section we give a first exposure and motivation for the relevance of category theory to interaction computing. We will briefly outline how regular sequential systems, such as the automata presented above, their more sophisticated version, i.e. timed automata as used by modelling services in BIONETS [70,71,72,73], and non-linear systems, represented by differential equations, can be generalised by means of category theory.

To show the relevance of category theory and explain to a computer scientist how it becomes a powerful translation machine, we outline which relationships have to be exploited in order to transform between different instances of automata described in different computational models. For this purpose, we first define an abstract transition system. Although its definition is very similar to the notion of automata introduced above, we do not call them automata because we want to avoid any confusion with the class of regular automata. A transition system is defined by a tuple $(\Sigma, S, s_0, \lambda)$, where Σ is the alphabet, S is the set of states, s_0 denotes the initial states, and λ is the transition relation $\lambda \subseteq S \times \Sigma \times S$.

With this general definition we can describe any automaton defined above in some monolithic form. However, if we consider a service composition in BIONETS we will realise that it consists of service individuals or service cells which interact with each other, concurrently, to generate complex services. Thus, modelling such a system also requires an abstraction into various transition systems which are able to interact. We take a similar approach as Winskel and Nielsen [74] who modelled concurrent processes with an eye on the calculus for communicating systems (CCS) developed by Milner [20]. From a BIONETS point of view, we can say that the way a service individual is communicating with another individual determines the relationship between them. For example, one individual may alter the execution flow by removing or adding certain branches in the flow. This can be seen as a type of atomic modification of the transition system. As a consequence, we could model such a communication between two services, or more precisely, between two transition systems T_1 and T_2 , as a transformation (σ, δ) , where σ denotes a function from the states of T_1 to those of system T_2 in which s_0 remains identical, i.e. $\sigma(s_0) = s_0$. δ is a partial function from the alphabet of T_1 to the alphabet of T_2 such that any transition in T_1 , (s, a, s') , is transformed into a transition $(\sigma(s), \delta(a), \sigma(s'))$ of T_2 if $\delta(a)$ is defined. If it is not defined then a kind of idle transition is defined for which $\sigma(s) = \sigma(s')$ holds. This allows the removal of certain transitions.

Finally, the transformation (σ, δ) forms a category \mathbf{T} in which the transition systems are the objects and (σ, δ) are morphisms on these objects. As a consequence, \mathbf{T} also supports important universal constructions, such as limits and co-limits, products, etc. (Sections 5 and 6 will provide more details). This again induces other important operations such as restrictions and re-labelling. Combinations of the latter and of universal properties can help to define parallel composition and nondeterministic sums. The definition of these operations directly induces a process language able to describe the parallel, potentially non-deterministic, execution of the service compositions described in BIONETS. In our case, this is not extremely interesting because we already know several languages which could have been used to describe such compositions. Thus, we only sketch this derivation process to show the potential of category theory and the status at which we arrived. In fact, if we derive a description from a biological system such as the p53 system, we will ultimately be able to define an execution model which reflects the particularities of the class of algorithms we are describing. In particular, we will need to find out how the universal properties in this category are defined and this in turn will help us to understand and define the required operations, such as composition and parallel execution. This basically results in a process category.

The concepts presented above were basically introduced in 1993. Their elaboration and extension include the definition of interaction categories by Abramsky [75]. Abramsky chose a new type of substitution and thereby introduced the notion of processes. Thus, in interaction categories, objects are basically interpreted as interface specifications and processes are maps between these specifications. Abramsky additionally unified various methodologies, starting at non-well-founded sets – the origin of category theory – and ending at concurrency theory, and defined the interaction categories SProc and ASProc for synchronous and asynchronous processes. Cockett and Spooner showed that SProc and ASProc can be constructed by using systems of open maps [76] and can also be embedded in the category of processes. Finally, Worytkiewicz, followed yet another approach [77]. He slightly modified the substitutions above by considering a transition system as a type of graph which attaches semantic information to the edges of the graph, which basically represents a control flow graph. Worytkiewicz calls this system categorical transition system because the semantic information appears to be best organised in a category.

Why do we summarise these different attempts, here? We want to highlight that there is not one category which possibly arises from one particular automaton. As we have seen – SProc and ASProc are just two examples – the opposite is the case. The different aspects of the underlying computation systems have to be investigated and their characteristics are reflected in the appropriate category. Thus, the category we will derive from biological systems in the future may also fit in the class of process categories. So, the appropriate representation of these systems in a specification language will need more human expertise. The tools used for all the insights listed above are based on the concepts of bisimulation, open maps, adjunctions and monads, and Kleisli- as well as Eilenberg-Moore categories. This list does not only partially reflect our research agenda but also shows the links between categorical structures important for our research. Why is this?

Open maps can be interpreted as so-called path-lifting properties [78]. Here, paths represent computations, e.g. sequences of consecutive transitions or a partial order of events, respectively, depending on what kind of representation has been chosen. Assume a morphism $f : X \rightarrow Y$ which can transform these paths into other paths while preserving behaviour. f is called an open map if it preserves the labels on the path, and if a path can be extended in Y using f then this extension can be matched to an appropriate extension in X . Obviously, this model can be considered as a general definition of bisimulation which tries to relate states of a transition system which behave identically and are not distinguishable by an external observer. Thus, in our context open maps can be used as a more concrete definition for the general notion of bisimulation applied to different types or categories of transition systems.

Adjunctions are in particular relevant for our research as they on the one hand allow for the linkage between an automaton and its behaviour, and on the other hand they enforce a particular structure on the category they start from. This special structure is called a monad. In fact, every adjunction gives rise to a monad on the category the adjunction starts from. Monads are an important concept in formal languages as they are able to model effects [79] such as global state manipulation, exception handling, text parsing, iterations, invoke continuations, simple express types of sequential computation. It is helpful to draw an analogy between adjunctions and regular functions. Adjunctions can be seen as a pair of functors where

one functor between two categories is the inverse of the other; thus, the conditions they impose on the starting category are in some way analogous to the conditions a regular function needs to satisfy in order to be invertible.

Finally, the Kleisli and Eilenberg-Moore categories are very special categories as they are initial and terminal objects, respectively, of the categories of adjunctions. This is particularly interesting as due to these characteristics the two categories can be used to answer the question: Given a monad \mathbf{C} , can we construct an adjunction which gives rise to this monad? With these ingredients we can! In fact, if we review the above and extend our understanding of category theory and the categorical characteristics of the automata derived from biological systems, we may derive for certain monads that are linked to biological system the corresponding adjunction. This again gives rise to the category that, finally, describes the behaviour of the corresponding machine. In this way, category theory may pave the way for a better understanding of biological systems.

Assuming the understanding of this ‘behaviour category’ and its influence on the monad it was basically derived from, i.e. its actual implementation, we imagine to use another functor which translates this behaviour category into a specification category which can be used to build a specification language for the Interaction Machine described above. Additionally, and as discussed further in Sections 5 and 6, the concept of minimal realisation, which uses adjunctions to derive a realisation from a specific behaviour – in our case – from a behaviour specification of a system, can be used to actually guide the automatic generation of a system which actually implements this behaviour.

Before we discuss categories in more detail, we wish to give an idea of how algebra can help formalise the structural properties of automata in general, and of automata derived from cellular pathways in particular.

4 Groups

Although the latest insights on the relevance of group and semigroup theory to interaction computing can be found in [43], what we have discussed so far in this chapter has hopefully provided enough ‘circumstantial evidence’ to motivate an in-depth study of permutation groups and transformation semigroups. The problem, however, remains that these topics are very abstract and difficult to understand by non-mathematicians. As we did in [1], therefore, we provide another ‘tutorial’ on abstract algebra, this time on group theory, that will hopefully make it more accessible to applied interdisciplinary scientists interested in bio-computing.

To this end, this section gathers our analysis of the rotational symmetry group of the tetrahedron from the reports in which it has appeared [4,5,6,7], since it provides a unifying thread and relatively concrete example. We emphasise that these reports contain more elementary facts and background that may help in following the discussion here, but that could not be included to keep this already long article from becoming even longer.

4.1 Geometry

Visualisations Fig. 11 shows the rotational symmetry group of the tetrahedron, shown as the 12 rotations that leave the appearance of this solid invariant relative to fixed observer. By labelling each vertex of the tetrahedron with a number, each rotation can be seen to permute the order in which these labels appear. The figure shows 12 different orientations and 12 different permutations of the vertices. But the total number of permutations of a set of 4 elements is $4! = 24$. The 12 missing permutations correspond to the 12 possible reflections and products of reflections of the tetrahedron relative to its centroid. The complete set of 24 permutations of 4 elements, isomorphic to the 24 symmetries of the tetrahedron, is called the **symmetric group** and since it is acting on 4 elements it is denoted by S_4 . We should not confuse the *symmetric* group, the group of all possible permutations of a set, with the *symmetry* group of a particular figure or set of elements, since the latter could be a subset, as we saw for the rotational symmetry group. Subsets of the symmetric group S_n (for any subscript n) are simply called permutation groups.

Fig. 12 shows the deterministic finite automaton (DFA) obtained from the rotational symmetry group of the tetrahedron. The states of this DFA are numbered to make it easier to see the correspondence with

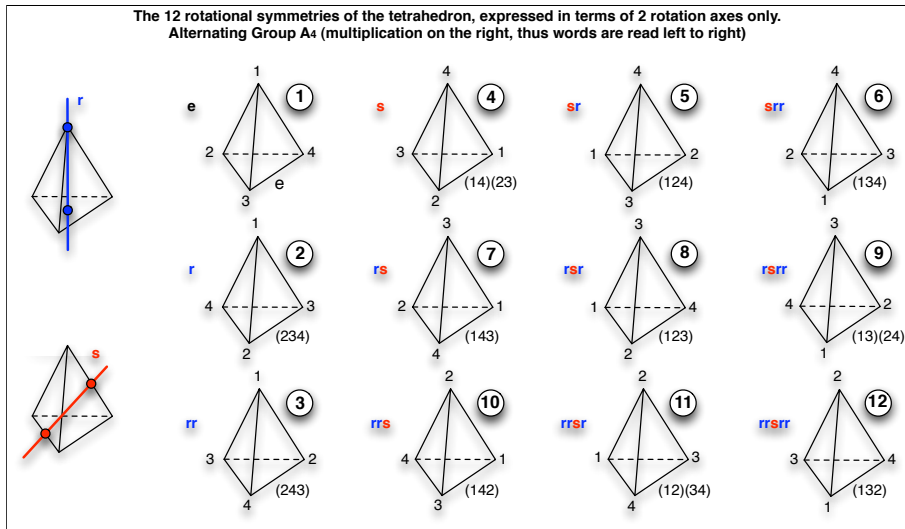


Fig. 11. Geometrical visualisation of the rotational symmetry group of the tetrahedron

the orientations of the tetrahedron shown in Fig. 11, where the same numbers are shown. We left the first state labelled as ‘e’ for consistency with Fig. 11, but it could also be labelled ‘1’. The periodic structure of this simple DFA is very evident, and can in fact best be appreciated in 3D, as shown in Fig. 13. The mathematical structure of this automaton is actually quite rich, as will be discussed in the sections below.

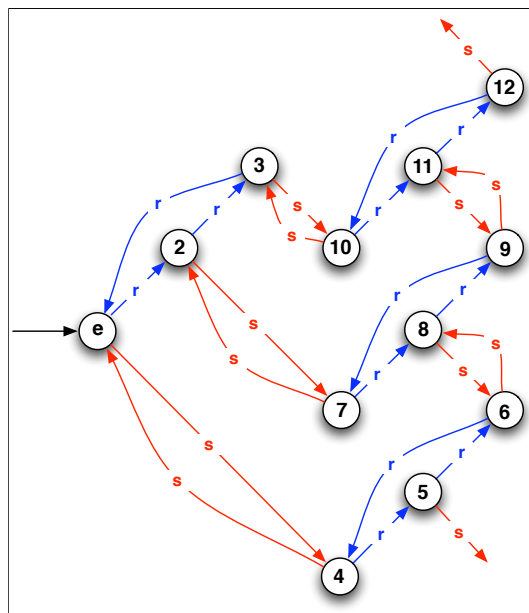


Fig. 12. DFA derived from the rotational symmetry group of the tetrahedron

The instantiation of the closure property of a symmetry group in the geometrical context of the tetrahedron exposes the fact that there are distinctly different ways in which this figure can be rotated while still satisfying the definition of a symmetry given above. Specifically, these are the rotations around the different axes of symmetry of the tetrahedron, which are shown as *r* and *s* in Figure 11. These rotations are called the *generators* because all the other elements of the group can be generated from them. The number of generators and how they generate the whole group is of fundamental importance for determining the structural properties of the group, a fact that is well exemplified by the generators of this group. The geometrically distinct character of the *r* and *s* rotations, in fact, corresponds to these two

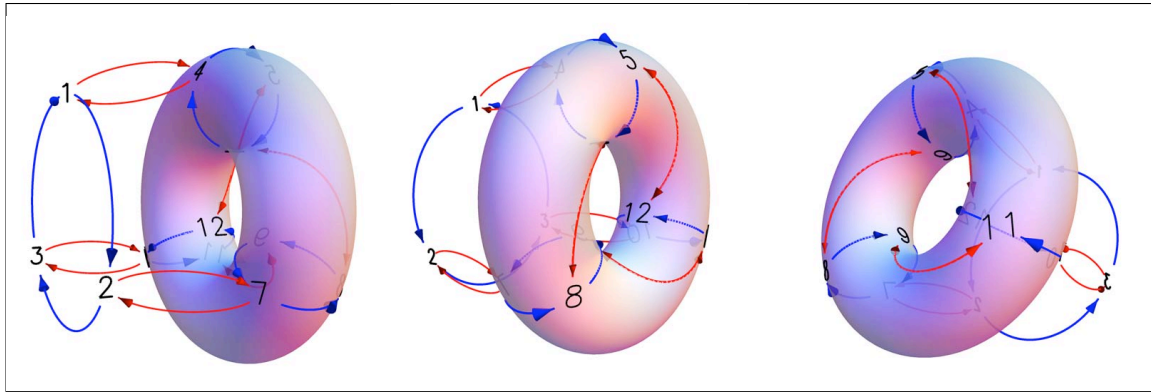


Fig. 13. Tetrahedron DFA in 3D

symmetries being the generators of their own distinct *subgroups*. But the story is a little more involved. Luckily, a more general formal machinery is available.

Symmetry levels map to subgroups We know that the 12 rotational symmetries of the group A_4 , by definition, leave the orientation of the tetrahedron invariant. But some of the transformations are ‘more symmetrical’ than others. For example the transformations e, r, rr all leave the position of the top vertex of the tetrahedron unchanged. Since, as we have said previously, the invertible transformations that leave some feature of a mathematical object invariant form a group, we might expect these three transformations to form a group, and this is in fact the case. Not only do they preserve the orientation of the tetrahedron, as the other 9 do, but in addition they also leave Vertex 1 unchanged. This additional level of symmetry is captured by the fact that these three elements of A_4 form a subgroup of A_4 (isomorphic to C_3).

In the parlance of permutation groups we say that Point 1 in the set $\Omega = \{1, 2, 3, 4\}$ is fixed by a subgroup of A_4 . This subgroup is called the stabiliser of Point 1 and it is denoted by G_1 (where in this case $G = A_4$). The next few paragraphs follow [80] fairly closely. Given a group G acting on a set Ω , denoted (Ω, G) , a point $\alpha \in \Omega$ is moved by elements of G to other points. The action of an element $g \in G$ on an element $\alpha \in \Omega$ is by right multiplication, and using the group-theoretical notation it is written αg . The set of images of α , over all the elements of G , is called the **orbit** of α under G :

$$\alpha^G = \{\alpha g : g \in G\}. \tag{5}$$

A sort of dual role is played by the elements of G that leave α fixed, called the **stabiliser** of α :

$$G_\alpha = \{g \in G : \alpha g = \alpha\}. \tag{6}$$

This is actually the *point* stabiliser since it fixes a single point of Ω .

Assertion 1. Two orbits are either equal or disjoint, i.e. each forms an equivalence class and together all the orbits partition Ω .

Proof: Let $\delta \in \alpha^G$ (so, necessarily, $\delta \in \Omega$). Then, $\delta = \alpha u$, for some $u \in G$. Now, we can construct the orbit of δ as $\delta^G = \{\delta x : x \in G\}$. But $\delta = \alpha u$; so, $\delta^G = \{\alpha u x : x \in G\}$. Since $u x$ will map to all the elements of G as x cycles over all the elements of G , $\delta^G = \alpha^G$. Therefore, if α^G and β^G have a common element δ , then $\alpha^G = \delta^G = \beta^G$. ■

Assertion 2. G_α is a subgroup of G .

Proof: Associativity is automatically satisfied since $G_\alpha \subseteq G$. $e \in G_\alpha$, since $\alpha e = \alpha$ (Identity). If $g \in G_\alpha$ and $h \in G_\alpha$, then $\alpha gh = \alpha h = \alpha$. Therefore, $gh \in G_\alpha$, and the same applies to hg (Closure). If $g \in G_\alpha$, $\alpha g = \alpha$. Now multiply both sides by g^{-1} : $\alpha g g^{-1} = \alpha g^{-1}$; then, $\alpha = \alpha g^{-1}$. Therefore, $g^{-1} \in G_\alpha$ (Inverse). ■

Assertion 3 (Orbit-Stabiliser Theorem). For any $x, y \in G$,

$$\alpha x = \alpha y \iff G_\alpha x = G_\alpha y. \tag{7}$$

This is saying that, if two different elements $x, y \in G$ map the same point $\alpha \in \Omega$ to the same image $\beta \in \Omega$, then x and y belong to the same coset of the subgroup G_α . This can be seen as a generalisation of the concept of stabiliser. In fact,

if $x, y \in G_\alpha$, then $\alpha x = \alpha y = \alpha$. On the other hand, even if we still have that $\alpha x = \alpha y$, but they equal a different point β , then the claim is that in this case x and y must be in the same coset of the stabiliser of α .

Proof: Since $\alpha x = \alpha y$, $\alpha xy^{-1} = \alpha$. Therefore, $xy^{-1} \in G_\alpha$. But this is the condition that ensures that x and y belong to the same (right) coset of the subgroup G_α ([4]: 40). As a consequence, $G_\alpha x = G_\alpha y$. The reverse proof is trivial, just follow the above steps in reverse. ■

Since the totality of the points of Ω to which α is mapped by the elements of G is the orbit of α , we have proven that there is a 1-1 correspondence between the elements of the orbit of α and the right cosets of G_α . Fig. 14 shows this fact for the group A_4 acting on the set of points $\{1, 2, 3, 4\}$, in particular when $\alpha = 1$. In this case there is only one orbit, so we say that G acts **transitively** on Ω .

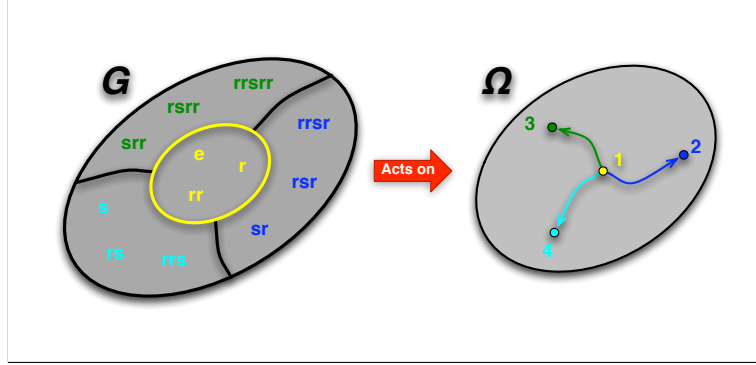


Fig. 14. Graphical visualisation of the stabiliser $G_1 = \{e, r, rr\}$ of the point $1 \in \Omega$ and of its right cosets, given $G = A_4$ and $\Omega = \{1, 2, 3, 4\}$. The action of G on Ω is transitive, i.e. there is only one orbit: given any starting point $\alpha \in \Omega$ the elements of G will map it to all the other points in Ω .

We need one more result that we will use below.

Assertion 4. $G_\beta = g^{-1}G_\alpha g$ when $\beta = \alpha g$.

Proof: Let $y \in G_\beta$; then, $\beta y = \beta$; $\alpha g y = \alpha g$; $\alpha g y g^{-1} = \alpha$. Therefore, $g y g^{-1} \in G_\alpha$. Since this can be done for any $y \in G_\beta$, we can write $g G_\beta g^{-1} \subseteq G_\alpha$, or: $G_\beta \subseteq g^{-1}G_\alpha g$. Now repeat the argument but start with $x \in G_\alpha$, leading to $g^{-1}G_\alpha g \subseteq G_\beta$. Since G_β is both a subset and a superset of $g^{-1}G_\alpha g$, we must have

$$G_\beta = g^{-1}G_\alpha g \quad (8)$$

■

The next subsections discuss Cayley's theorem to build our intuition of the action of permutation groups, which we will need to conceptualised how they can act on sets of states.

4.2 Groups as sets of functions

In calculus and analytic geometry we routinely talk about functions over the reals, using notation such as

$$f : \mathbb{R} \rightarrow \mathbb{R}. \quad (9)$$

An example of such a function could be $y = f(x) = x^3$, $x, y \in \mathbb{R}$, with its familiar graph. Given the very different context of abstract algebra over discrete finite sets, it was not immediately apparent that a permutation group G operating on a set Ω can be understood in a conceptually *identical* way:

$$G : \Omega \rightarrow \Omega, \quad (10)$$

where, however, G is actually a set of (invertible) functions rather than a single function f . To make the analogy also formally identical we only need to take an element of G at a time:

$$g : \Omega \rightarrow \Omega, \quad g \in G. \quad (11)$$

In other words, a particular permutation of, for example, a set of 4 numbers is no different conceptually to a function that maps the real axis to itself according to the rule x^3 . Furthermore, we can graph it in the same way. Take for instance $\Omega = \{1, 2, 3, 4\}$, $g = (124)$, and $g \in G = A_4$. The graph of this function operating on the finite and discrete set Ω is shown in Fig. 15. The lines connecting the dots are only meant as a visual aid, the dotted line representing the identity, and the axes labels are different to what we would normally see on the graph of a real function. Normally we would label the axes x and y , whereas the analogue of the labels shown in this figure would be \mathbb{R} on both axes. It is interesting to note that working with discrete sets makes the invertibility of the functions in some sense ‘easier’ to attain. In fact, a continuous curve going through the origin and these 4 points, which could be a cubic, is not invertible because it is not 1-1, whereas $g = (124)$ is. As long as two red dots are not at exactly the same y -coordinate, in fact, a discrete function such as this is invertible.

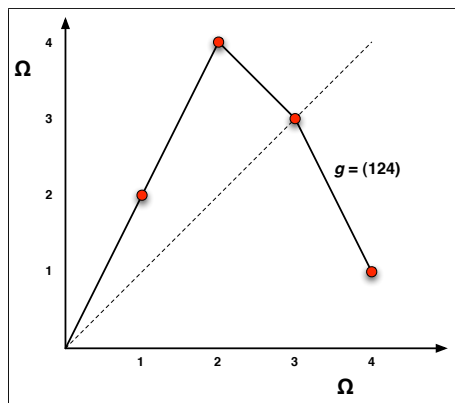


Fig. 15. Graph of the permutation $g = (124)$ of a set of 4 elements

Whereas each function $g \in G$ can be seen to map each point $\alpha \in \Omega$ to another point $\beta \in \Omega$,¹² strictly speaking a permutation is a function that acts on all the points of Ω *simultaneously*. Formally, therefore, the more correct representation of a permutation is as a vector function:

$$g : \Omega^n \rightarrow \Omega^n, \quad g \in G, \quad (12)$$

where n is the size of Ω . We have not bothered to change the typeface of g to indicate vector character because normally the intended interpretation is clear from the context. As we discuss below these two representations correspond to sequential and parallel execution, respectively, when Ω is the set of states of an automaton.

As shown in Fig. 16, this view of groups as sets of functions then facilitates the conceptualisation of group actions, which are defined formally as

$$\mu : G \times \Omega \rightarrow \Omega, \quad (13)$$

and that can therefore be visualised as a surface.

Compared to applied analysis, which merely classifies functions in terms of the governing differential equations they solve, without attempting to relate all these functions to each other, the richness and detail of the structure of finite groups of permutations that has been uncovered since the time of Galois is truly staggering. Here ‘structure’ refers to the character, properties, and topology of the *relationships* between the elements of groups of permutations. Whereas in applied analysis we deal with individual functions (e.g. e^x , $\tanh(x)$, $\sin(x)$, etc.) operating on the real (or complex) numbers, in abstract algebra we deal with *sets* of functions. The individual permutations are almost irrelevant. The important cognitive structures are much more complex and nested, such as groups, algebras, etc. and they operate on different kinds of objects, such as rings, fields, groups, algebras, partial orders, etc. From the point of view of reflecting

¹² $g(4) = 1$ in the figure above, just like $f(2) = 8$ in the case of the cubic.

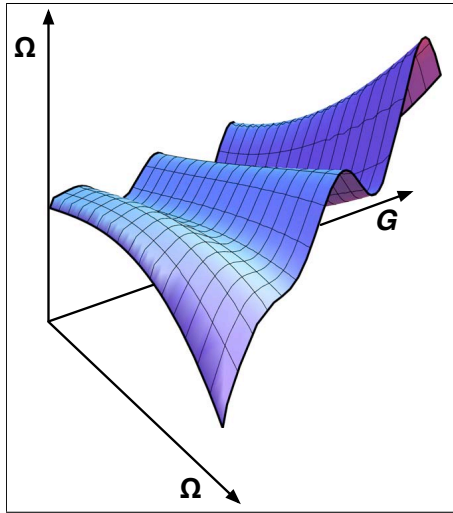


Fig. 16. Visualisation of group action

the complexity of the real world, and of biological systems in particular, abstract algebra is immensely richer and more powerful. However, its abstractness makes it challenging to add sufficient detail to make algebra an effective modelling tool for many applied problems, especially those that rely on physical processes.

With this context, we can better appreciate the significance of Lie’s work in inventing Lie groups, which exhibit algebraic properties whilst being themselves representable as continuous manifolds over the reals. This integration of algebra and analysis was the great achievement of Sophus Lie in developing his theory ([81], viii). Lie groups unify the solution of linear/non-linear, ordinary/partial differential equations. Table 2 summarises the corresponding roles of mathematical structures relevant to discrete and continuous mathematics.

	Discrete	Continuous
Group	G	Lie group as symmetry (map): $(x^*, y^*) = f(x, y; \epsilon), \forall \epsilon \in \mathbb{R}$
Group G operates on	Ω	Solution curve of ODE
Right Regular Representation	(G, G)	Symmetry mapping points along its integral curve

Table 2. Comparison of analogous discrete and continuous algebraic concepts

4.3 Cayley’s theorem

From symmetries to permutations The two interpretations of g as scalar or vector function map to the interpretation of its action as a symmetry or a permutation, respectively. In the symmetry group interpretation, we can apply any of the rotations (expressed as products of r and s symmetries) to any of the 12 orientations of the tetrahedron, and we will obtain another orientation from the same set of 12 (the group is closed). However, in the permutation group interpretation, we know from our discussion of Eq. (12) that each element of G operates on the *whole* set at once, in parallel.

The only possible way to make the physical discussion ‘keep up’ with the abstract discussion is to imagine that we are not dealing with a single tetrahedron that can be rotated 12 different ways, but that, in fact, we are dealing with **12 tetrahedra** that can simultaneously be spun around their symmetry axes, simultaneously and in parallel, in the manner we have described. Hence the physical realisation of a permutation group is necessarily an ensemble of parallel systems.

From symmetries to the right regular representation We now show that the action of a group G on the elements of a set Ω is entirely analogous to the action of the group elements on each other. We do this by demonstrating that the closure property can be expressed in more abstract terms, such that Ω can be forgotten and the action of G can be kept within G .

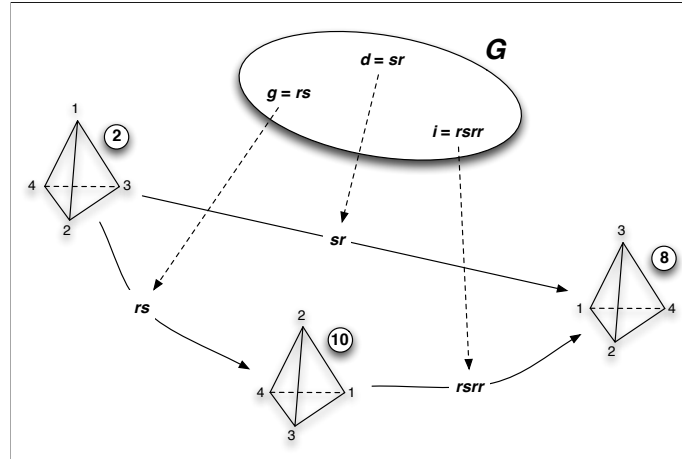


Fig. 17. Closure of group action on a set of states Ω (only 3 of the 12 states shown)

Fig. 17 shows that applying transformation rs to state 2 to obtain state 10 followed by transformation $rsrr$ to obtain state 8 is equivalent to applying transformation sr to state 2 to obtain state 8 directly, all three being transformations that belong to the same group G . If we use the notation of Table 7 to label the group elements, we can write this as follows:

$$g(2) = 10 \tag{14}$$

$$i(10) = 8. \tag{15}$$

These two transformations can be composed:

$$i(g(2)) = 8 \tag{16}$$

and, as the figure shows, they are equal to a third from the same group,

$$d(2) = 8. \tag{17}$$

We can now equate Eqs. (16) and (17), obtaining

$$i(g(2)) = d(2). \tag{18}$$

But because of closure this statement holds irrespective of the starting state. Hence,

$$i(g) = d. \tag{19}$$

Using the notation of multiplication on the right,

$$gi = d, \tag{20}$$

which shows that a group element can act on another group element to obtain a third. In this manner we have lifted the group operation from a *rotation* to the more abstract form of *functional composition*. Letting all the group elements act on themselves, we reach the right regular representation of the group acting on itself, (G, G) .¹³

¹³ Please see Appendix B of [5] for the explanation of a potentially confusing technical point.

We can now better appreciate and understand how the Cayley diagram shown in Fig. 18 should be interpreted (the labels shown in the figure are consistent with Table 3). Rather than a static structure that we traverse a step at a time to reach a particular state, a Cayley diagram is best interpreted as a wholly dynamic object. The application of each element of the group, for example of each generator, causes *all* the balls to shift position, simultaneously. This is not necessarily obvious from the figure above because the balls are all identical. But if we were to paint the labels a, b, c, \dots shown in the figure on the balls themselves we would observe *all* the balls changing position simultaneously each time an r or an s transformation is applied.

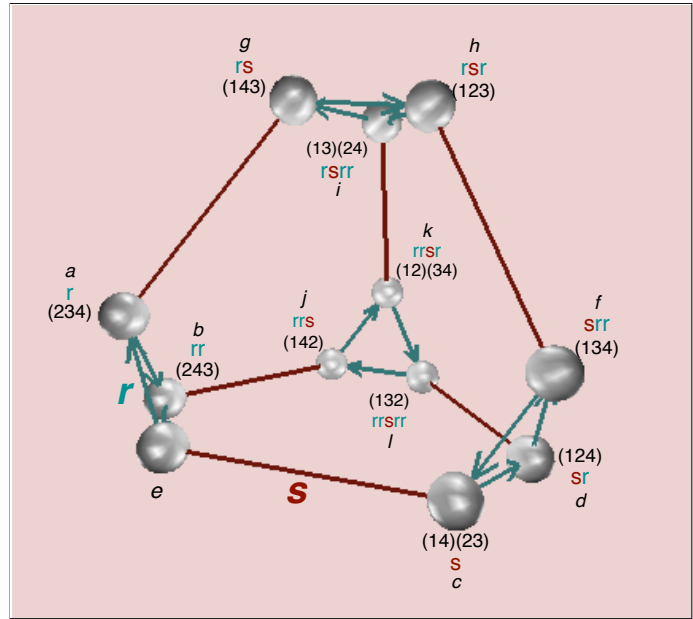


Fig. 18. Cayley diagram for the group A_4 [82]

Symmetry	e	r	rr	s	sr	srr	rs	rsr	$rsrr$	rfs	$rfsr$	$rfsrr$
State	1	2	3	4	5	6	7	8	9	10	11	12
Permutation (G)	e	(234)	(243)	(14)(23)	(124)	(134)	(143)	(123)	(13)(24)	(142)	(12)(34)	(132)
Vertices (Ω)	1234	1423	1342	4321	4132	4213	3241	3124	3412	2431	2143	2314
Label (G)	e	a	b	c	d	f	g	h	i	j	k	l

Table 3. Labelling the group elements of the tetrahedron’s rotational symmetry group

If each ball is regarded as filling a ‘hole’, whose position relative to the page on which the figure is drawn is fixed as the balls move around pushed by the various group elements, then each slot can be associated with a specific, and fixed, bit position in a 12-bit word: for example, $(e, a, b, c, d, f, g, h, i, j, k, l)$, or $(g, h, i, a, b, e, f, c, d, k, l, j)$, etc. Hence, each time we operate on the set of balls with a group element, we are permuting this set. The set of 12 permutations that correspond to the 12 orientations of the tetrahedron is shown in Table 4. If we are not constrained by the geometry of the tetrahedron, how many permutations are there? If this were a 12-bit word over a 12-symbol alphabet the answer would be $12^{12} = 8,916,100,448,256$. However, with permutations we can’t have repeated symbols in any one word, so that cuts down the number of possibilities to $12! = 479,001,600$, of which here we are discussing only 12. These 12 permutations happen to form a group, since any combination thereof yields another permutation from the same set of 12, and never one from the remaining 479,001,588. The set of the $12!$ possible permutations of 12 elements is called the Symmetric group of 12 elements and is denoted by S_{12} .

	<i>e</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
<i>e</i>	<i>e</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
<i>a</i>	<i>a</i>	<i>b</i>	<i>e</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>c</i>	<i>d</i>	<i>f</i>
<i>b</i>	<i>b</i>	<i>e</i>	<i>a</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
<i>c</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>e</i>	<i>a</i>	<i>b</i>	<i>l</i>	<i>j</i>	<i>k</i>	<i>h</i>	<i>i</i>	<i>g</i>
<i>d</i>	<i>d</i>	<i>f</i>	<i>c</i>	<i>l</i>	<i>j</i>	<i>k</i>	<i>h</i>	<i>i</i>	<i>g</i>	<i>e</i>	<i>a</i>	<i>b</i>
<i>f</i>	<i>f</i>	<i>c</i>	<i>d</i>	<i>h</i>	<i>i</i>	<i>g</i>	<i>e</i>	<i>a</i>	<i>b</i>	<i>l</i>	<i>j</i>	<i>k</i>
<i>g</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>a</i>	<i>b</i>	<i>e</i>	<i>f</i>	<i>c</i>	<i>d</i>	<i>k</i>	<i>l</i>	<i>j</i>
<i>h</i>	<i>h</i>	<i>i</i>	<i>g</i>	<i>f</i>	<i>c</i>	<i>d</i>	<i>k</i>	<i>l</i>	<i>j</i>	<i>a</i>	<i>b</i>	<i>e</i>
<i>i</i>	<i>i</i>	<i>g</i>	<i>h</i>	<i>k</i>	<i>l</i>	<i>j</i>	<i>a</i>	<i>b</i>	<i>e</i>	<i>f</i>	<i>c</i>	<i>d</i>
<i>j</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>b</i>	<i>e</i>	<i>a</i>	<i>i</i>	<i>g</i>	<i>h</i>	<i>d</i>	<i>f</i>	<i>c</i>
<i>k</i>	<i>k</i>	<i>l</i>	<i>j</i>	<i>i</i>	<i>g</i>	<i>h</i>	<i>d</i>	<i>f</i>	<i>c</i>	<i>b</i>	<i>e</i>	<i>a</i>
<i>l</i>	<i>l</i>	<i>j</i>	<i>k</i>	<i>d</i>	<i>f</i>	<i>c</i>	<i>b</i>	<i>e</i>	<i>a</i>	<i>i</i>	<i>g</i>	<i>h</i>

Table 4. Cayley table for A_4 , using the tetrahedron’s rotational symmetry group elements. As explained very clearly in [83], the colour code used in this table and in Table 3 corresponds to the stabiliser G_1 and its left cosets.

With this, we have effectively proved Cayley’s theorem, although in an intuitive rather than formal way:

Cayley’s theorem. Every group G is isomorphic to a subgroup of the symmetric group S_G acting on its own elements.

4.4 Symmetry types map to permutation representations

A similar story to the development of Section 4.1 can be told by focussing more on the set Ω than on the group G that acts on it. In particular, the 12 rotational symmetries of the tetrahedron can be seen to preserve different geometrical features of the tetrahedron:

- the 12 configurations of the tetrahedron (obtainable by rotations)
- the 6 edges
- the 4 vertices or the 4 faces
- the 3 diagonals, where by ‘diagonal’ we mean an axis that bisects two opposite edges, such as the symmetry axis s in Fig. 11

We can describe the invariance of these different geometrical features as different *types* of symmetry originating from the same group. At the same time, we notice that each of these features can be seen as different sets of objects that are permuted by the same abstract group G . This situation has been formalised by generalising the concept of permutation group to a ‘permutation representation’, and Cayley’s theorem to a more general construction. For each finite abstract group there are an infinite number of permutation representations, subdivided into a finite number of separate isomorphic classes. If we stick with the tetrahedron it is sufficient to show the 4 representations of A_4 to demonstrate the general idea.

The statement of Cayley’s theorem above can be restated as (see Fig. 19):

Cayley’s theorem. Given any group G , there exists an injective (or faithful) homomorphism from G to S_G , $\phi : G \rightarrow S_G$ (or, equivalently, $\phi : G \rightarrow S_{|G|}$)

This is what we now generalise. In the following, G is an abstract group, Ω is the set G operates on, as before, and ϕ indicates a group homomorphism.

Definition 5. A **permutation representation** of G on Ω is a homomorphism from G to S_Ω , $\phi : G \rightarrow S_\Omega$.

We denote the image of the homomorphism by $\text{Im}(\phi)$ and we call it K , whereas the kernel is denoted by $\text{Ker}(\phi)$ (see [4]: 43). The First Isomorphism Theorem then gives us that

$$K \cong G/\text{Ker}(\phi), \tag{21}$$

where \cong means “isomorphic to”. With this, we can now denote a permutation representation as (Ω, K) . Fig. 20 shows how a particular permutation group can be obtained from the abstract group it corresponds to.

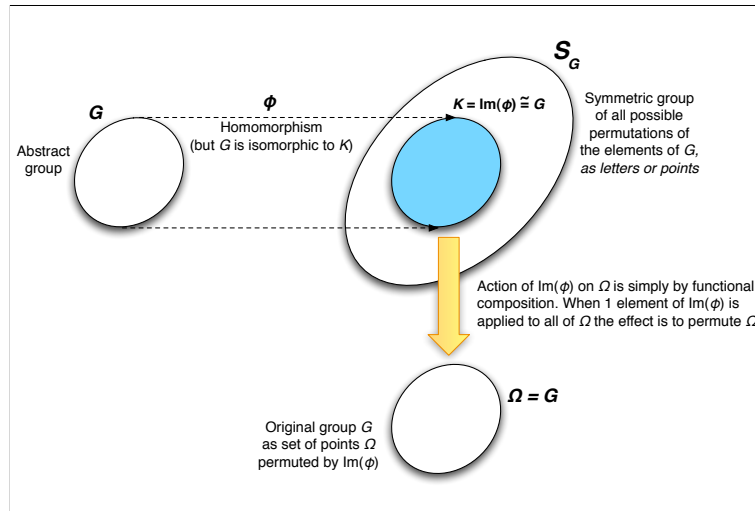


Fig. 19. Graphical visualisation of Cayley’s theorem

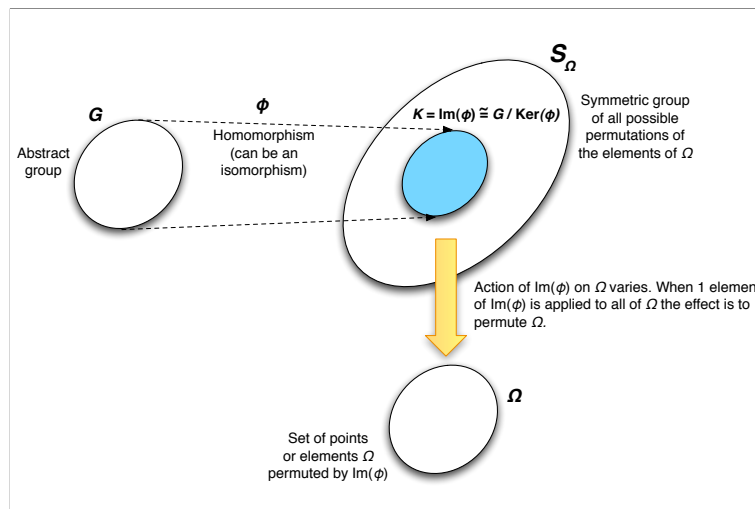


Fig. 20. Graphical visualisation of the relationship between an abstract group and one of the permutation groups derivable from it

Now let’s add some structure, first with a somewhat informal discussion. Given any *abstract* group G and any one of its subgroups H , we can in general define the action of G on the (right) cosets of H by right multiplication. G together with this action and $\Omega = \{\text{the set of (right) cosets of } H\}$ can be seen as a permutation group. Therefore, H is necessarily and automatically the stabiliser of this action, G_H .

Second, we presume that G is acting transitively as a *permutation* group on a set of points Ω . Taking any one of these points, say α , we have seen that its stabiliser G_α is a subgroup of G . Therefore, it must be one of the subgroups we have already examined in the first step. The orbit-stabiliser theorem tells us that the action of G on Ω (e.g. the orbit) is isomorphic to the action of G on the cosets of G_α .

As a consequence, as a third step we reach the surprising conclusion that, **in order to see how an abstract group G can be understood as a transitive permutation group acting on any set Ω , we simply need to look at the action of G on the (right) cosets of one of its subgroups.**¹⁴

Before we can apply the above concepts to the systematic analysis of a given group, we need a few more concepts; in particular, a method to determine the kernel of a group homomorphism. This can be done through one more construction, the ‘core’ of a subgroup ([84]: 4).

¹⁴ This is the generalisation of Cayley’s theorem mentioned above, and restated more carefully below.

Definition 6. The core of a subgroup $H \leq G$ is the intersection of all the subgroups that are conjugate to H :

$$\text{Core}_G(H) = \bigcap_{g \in G} g^{-1}Hg \tag{22}$$

Assertion 7. When Ω is the set of cosets of $H \leq G$, the kernel of a group homomorphism $\phi : G \rightarrow S_\Omega$ that represents the action of G on Ω equals the core: $\text{Ker}(\phi) = \text{Core}_G(H)$.

Proof: In order to understand Eq. (22) and to prove that it does indeed equal the kernel of the homomorphism we need to use the terminology and formalism we have already defined a little more carefully than we have done above, relying on Fig. 21. We begin with the stabilisers of the coset H and of any other coset Hg :

$$K_H = \{x\phi : H(x\phi) = H\} = H\phi \tag{23}$$

$$K_{Hg} = \{x\phi : Hg(x\phi) = Hg\} = (g^{-1}Hg)\phi \tag{24}$$

Now we recall that the purpose of a permutation representation through a homomorphism is to transfer the properties of an abstract group G to a permutation group K acting on a particular set Ω , in such a way that the group properties are preserved. For instance, the (transitive) action of K on Ω is closed in the sense that, for $h \in K$ and $\alpha \in \Omega$, there exists a $\beta \in \Omega$ such that $\alpha h = \beta$. Similarly, there is a unique identity element $e_K \in K$ that does not move any of the points of Ω .

The uniqueness of e_K means that the intersection of all the stabilisers of Ω in K cannot contain more than e_K . Then we note that H seen as a subgroup of G is (trivially) the stabiliser of H seen as a point of $\Omega = \{\text{set of (right) cosets of } H\}$. Using the orbit-stabiliser theorem and Eq. 8, we can see that $g^{-1}Hg$, for all $g \in G$, are the stabilisers of the points $Hg \in \Omega$. With that, we can finally write:

$$\begin{aligned} \text{Ker}(\phi) &= \{x \in G : x\phi = e_K\} = e_K\phi^{-1} = \left(\bigcap_{\alpha \in \Omega} K_\alpha \right) \phi^{-1} = \left(\bigcap_{g \in G} K_{Hg} \right) \phi^{-1} \\ &= \left(\bigcap_{g \in G} (g^{-1}Hg)\phi \right) \phi^{-1} = \bigcap_{g \in G} ((g^{-1}Hg)\phi)\phi^{-1} = \bigcap_{g \in G} g^{-1}Hg = \text{Core}_G(H). \end{aligned} \tag{25}$$

■

Assertion 8. $\text{Core}_G(H)$ is the largest normal subgroup of G contained in H .

Proof: We know from the First Isomorphism Theorem that the kernel of a homomorphism is a normal subgroup of the domain group (also called the preimage). Thus the core is normal. If it is the meet (intersection) of all the conjugates of H it must be in H . Now let N be the largest normal subgroup of G contained in H , $N \subseteq H$. Then, $N = g^{-1}Ng \subseteq g^{-1}Hg$. But since this is true for any $g \in G$, $N \subseteq \bigcap_{g \in G} g^{-1}Hg$. Thus, $N \subseteq \text{Core}_G(H)$. But since the core is normal, $N = \text{Core}_G(H)$. ■

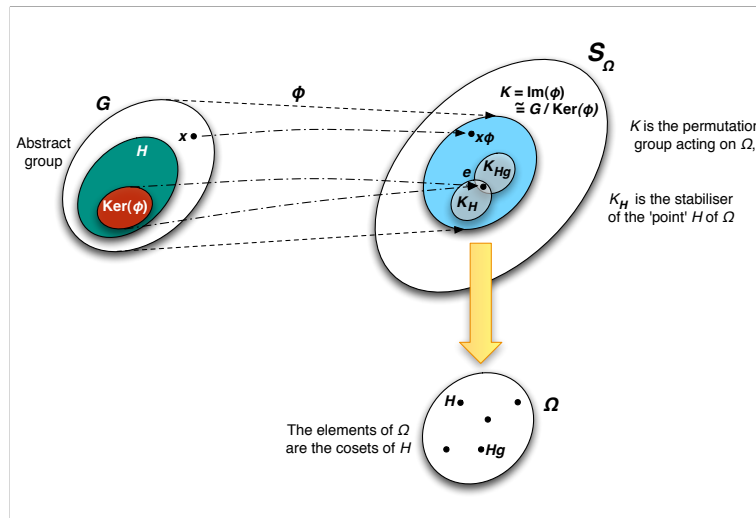


Fig. 21. Visualisation in support of the derivation of the kernel

Finally, we need to generalise the definition of point stabiliser. As we might expect, sets of invertible transformations that preserve *subsets* of points $\in \Omega$ also generate subgroups of G . It turns out that there are two ways to do that.

Definition 9. Given a subset $\Delta \subseteq \Omega$, the **pointwise stabiliser** is the set of elements of G that fix *each and every* point of Δ :

$$G_{(\Delta)} = \{g \in G : \delta g = \delta, \quad \forall \delta \in \Delta\}. \tag{26}$$

Definition 10. Given a subset $\Delta \subseteq \Omega$, the **setwise stabiliser** is the set of elements of G that map the element of Δ to themselves:

$$G_{\{\Delta\}} = \{g \in G : \Delta g = \Delta\}. \tag{27}$$

Interestingly, in general the pointwise stabiliser is a normal subgroup of the setwise stabiliser, although we will not prove this assertion. In any case the fact that these two stabilisers are also subgroups of G is important because it means that the orbit-stabiliser theorem applies also to them.

Case	Subgroup H , i.e. stabiliser K_α	$ \Omega $, as set of cosets of H	Representation of permutation group of the cosets of H
1	$\{e\}$	12	Cayley (or right regular) representation: A_4 acts on itself by right multiplication
2	$\{e, (14)(23)\}$	6	In this case H has 6 cosets
3	$\{e, (234), (324)\}$	4	Familiar representation of A_4 acting on 4 elements: H has 4 cosets
4	$\{e, (12)(34), (13)(24), (14)(23)\}$	3	Cyclic over 3 elements: H is normal and has 3 cosets

Table 5. Permutation representations induced by different subgroups of A_4

With the above, we finally have enough machinery to perform a systematic analysis of A_4 . Table 5 shows the 4 cases we need to examine. Table 6 summarises the results of the analysis of A_4 .¹⁵ Let’s go through each case in turn:

- In Case 1, the stabiliser of any given configuration of the tetrahedron can only be the identity transformation. Therefore, the cosets of the identity element are all the other elements, and Ω has size 12 (this is called the **degree** of the permutation group). This situation is the familiar one where each transformation can also be regarded as a state. If we characterise a configuration of the tetrahedron by its four vertices, then $\Delta = \Omega = \{1, 2, 3, 4\}$. Then we see that the pointwise stabiliser is $\{e\}$, whereas the setwise stabiliser is $G = A_4$.
- In Case 2 we are looking at one edge of the tetrahedron at a time, and the transformation that belongs to the stabiliser of this object flips it over: $(14)(23)$ for edge 1-4, for instance. Thus, in this case $\Delta = \{1, 4\}$ and we are talking about its setwise stabiliser: $H = \{e, (14)(23)\}$, so it will have 6 cosets in all, including itself. The reason there are only 3 conjugates instead of 6 is that e.g. the transformation that operates on (14) is the same that operates on (23) (and this happens to the two other pairs of opposite edges as well). Therefore, when it is (23) ’s turn to be flipped it will give rise to the same stabiliser as (14) ’s. Hence there will be 3 pairs of identical stabilisers instead of 6, and that’s why we have 3 conjugate subgroups instead of 6 (for a visualisation of the subgroup lattice see [4]: 36). Because these three subgroups intersect only on the identity, $\text{Ker}(\phi) = \{e\}$ and A_4 acting on these cosets is isomorphic to A_4 acting on $\Omega = \{6 \text{ edges of tetrahedron}\}$.
- In Case 3 the invariant is the location of a particular vertex. So the corresponding stabiliser is the subgroup that contains the rotations around an axis going through that vertex. Because there are 4 vertices, this action comes with 3 conjugates to whichever stabiliser one chooses first, corresponding to the other 3 vertices. The stabiliser is itself conjugate to these 3, so there are 4 mutually conjugate subgroups in all in this case. Because the 4 axes are pointing in different directions in space and do not cross at any of the vertices, clearly the 4 stabilisers only have the identity in common, which explains why $\text{Ker}(\phi) = \{e\}$. Since each stabiliser in this case has 4 cosets (including itself), this action is isomorphic to the abstract group acting on itself (Cayley representation).
- Case 4 is quite interesting because in this case the feature being stabilised are the three ‘diagonals’ simultaneously. This is subtle because permutation $(14)(23)$, for example, leaves the red diagonal in Table 6 unchanged, whereas it flips the other two by 180 degrees. This is similar to Case 2, where we saw that flipping is still a symmetry. If now the orientation of the three diagonals is changed by applying a group element so that, for instance, the red diagonal is pointing to the top left rather than the top right, subsequent application of the elements of the *original* stabiliser (such as $(14)(23)$) will *still* preserve the 3 diagonals. Therefore, the stabiliser is the same. This is equivalent to saying that the stabiliser subgroup is its own conjugate and, as we know, this means that this stabiliser is a normal subgroup. In particular, as shown H is isomorphic to the Klein group, denoted V_4 .

We are now ready for a more careful statement of the generalised Cayley’s theorem, which in a sense ‘bounces’ us back to abstract groups:

¹⁵ Web reference for the graphic in Case 1: <http://math.about.com/od/geometry/ss/platonic.htm>.

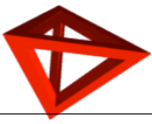
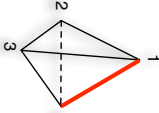
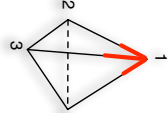
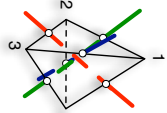
Geometrical invariant	H , or Stabiliser K_α	Homomorphism	$\text{Ker}(\phi)$	K	Number of conjugates of $K_\alpha \cong H$	Right cosets of H	Conjugate subgroups $g^{-1}Hg$
<div style="text-align: center;">  <p>1</p> </div>	$\{e\}$	$\phi: A_4 \rightarrow S_{12}$	$\{e\}$	$A_4/\{e\} \cong A_4$	1	$\{e\}$ $\{(234)\}$ $\{(243)\}$ $\{(14)(32)\}$ $\{(143)\}$ $\{(142)\}$ $\{(124)\}$ $\{(123)\}$ $\{(12)(34)\}$ $\{(134)\}$ $\{(13)(24)\}$ $\{(132)\}$	$\{e\}$
<div style="text-align: center;">  <p>2</p> </div>	$\{e, (14)(23)\} \cong C_2$	$\phi: A_4 \rightarrow S_6$	$\{e\}$	$A_4/\{e\} \cong A_4$	3	$\{e, (14)(23)\}$ $\{(234), (124)\}$ $\{(243), (134)\}$ $\{(143), (132)\}$ $\{(123), (142)\}$ $\{(12)(34), (13)(24)\}$	$\{e, (14)(23)\}$ $\{e, (12)(34)\}$ $\{e, (13)(24)\}$
<div style="text-align: center;">  <p>3</p> </div>	$\{e, (234), (324)\} \cong C_3$	$\phi: A_4 \rightarrow S_4$	$\{e\}$	$A_4/\{e\} \cong A_4$	4	$\{e, (234), (243)\}$ $\{(14)(32), (143), (142)\}$ $\{(124), (123), (12)(34)\}$ $\{(134), (13)(24), (132)\}$	$\{e, (234), (324)\}$ $\{e, (123), (132)\}$ $\{e, (134), (143)\}$ $\{e, (124), (142)\}$
<div style="text-align: center;">  <p>4</p> </div>	$\{e, (14)(23), (12)(34), (13)(24)\} \cong V_4$	$\phi: A_4 \rightarrow S_3$	V_4	$A_4/V_4 \cong C_3$	1	$\{e, (14)(23), (13)(24), (12)(34)\}$ $\{(234), (124), (143), (132)\}$ $\{(243), (134), (123), (142)\}$	$\{e, (14)(23), (13)(24), (12)(34)\}$

Table 6. Geometrical invariants of the tetrahedron and permutation representations of A_4

Generalised Cayley’s Theorem. Every transitive permutation group is isomorphic to an abstract group acting on its (right) cosets.

The deep implication we had hinted at at the beginning of this section is that, therefore, a new emphasis is placed on the algebraic structure of the instruction set that operates on a set of states. Hence, studying the algebraic structure of groups becomes a high priority in the development of a mathematical framework for computational spaces. This gives greater relevance than we previously suspected to the Jordan-Hölder and Krohn-Rhodes decomposition theorems for finite groups and semigroups, respectively. We can only give a flavour of the former by pointing out how every finite group can be decomposed into a finite ‘composition series’ of nested subgroups, where each is normal in the subgroup immediately above it. In the case of A_4 such a series is: $A_4 \supseteq V_4 \supseteq C_2 \supseteq \{e\}$. Thus, the stabiliser in Case 3, although it is a perfectly legitimate subgroup of A_4 (C_3), does not belong to any composition series of this particular group. Fig. 22 shows these facts graphically. The notation refers to partitions of states of the tetrahedron DFA and follows [4]; also shown are the graphical representations of the partitions, where the blocks refer to different subsets of states in each case. In case case, B_0 corresponds to the stabiliser H and the other blocks are its other cosets.

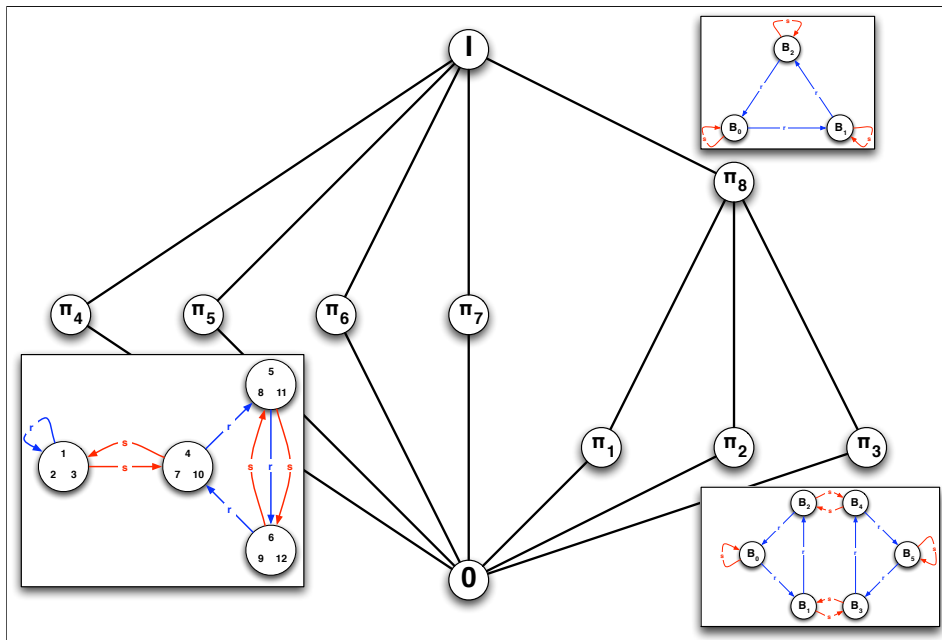


Fig. 22. Subgroup lattice of A_4 , showing the conjugate subgroups and the coset partitions at each level of the decomposition

Let’s spell out the implications a little more carefully. For each choice of geometrical feature we get a different stabiliser, and therefore we get a different permutation group: A_4 acting on 12, 6, and 4 elements in the first three cases, and C_3 acting on 3 elements in the fourth case. As we have proven in the preceding pages, there is an isomorphism between the ‘elements’ as geometrical features (or numbers, or whatever) and the cosets of the stabiliser in question. This is already very interesting, and leads to the generalisation of Cayley’s theorem stated above, but there is more. In some cases, namely when the stabiliser (e.g. V_4 in this case) is not just a subgroup but a *normal* subgroup of the group above it (A_4 in this case), the cosets acquire an additional property. They become the elements of a *new* group, the factor group (in this case, $A_4/V_4 = A_4/\text{Ker}(\phi) \cong K$).¹⁶

Then the same process repeats. In fact, although from the point of view of A_4 Case 2 is a relatively uninteresting permutation of six elements, with the stabiliser $\{e, (14)(23)\} \cong C_2$ not a normal subgroup of A_4 , the same subgroup *is* a normal subgroup of V_4 . Thus, it reproduces the same pattern again of a

¹⁶ As we just proved in Assertion 8 the kernel is the largest normal subgroup of G contained in H . Thus, if H is normal, then necessarily $\text{Ker}(\phi) = H$.

homomorphic image (C_2) isomorphic to a factor group (V_4/C_2) acting on the (two, in this case) cosets of the stabiliser (C_2), at yet another scale of description.

The reason this is fundamentally interesting is that at each level of the decomposition the same cosets can be seen as ‘states’ or objects being permuted and as elements of a new group doing the permuting. This is again the familiar theme of the Cayley or right regular representation, but now it is acting at different scales of description, which become progressively ‘coarser’. In this we recognise the nested pattern of biological systems, where the structural and functional roles of the components alternate and intermix in uncannily similar ways.

It is for this reason that we have striven to provide an in-depth discussion of the hierarchical structure of groups. With the above we have given an intuitive explanation of the Jordan-Hölder theorem, which is conceptually analogous to the Krohn-Rhodes theorem for semigroups. We now present a brief discussion of some basic semigroup concepts.

4.5 Semigroups acting on states

In our work on the topic of group theory we wanted to reach a detailed discussion of the Jordan-Hölder theorem on the decomposition of finite groups as a basis for a discussion of the Krohn-Rhodes prime decomposition theorem for semigroups [44], since the latter is relevant to the analysis of automata derived from cellular pathways [48,50,16]. As previously discussed [5,6], our interest is in understanding how the presence of groups in the Krohn-Rhodes decomposition of the semigroups associated with automata derived from cellular pathways could best be interpreted in terms of the behaviour and computational properties of the pathways being analysed. Such insights may help us reach more general conclusions about the computational (architectural and algorithmic) properties software systems need to have in order to exhibit self-organising behaviour similar to the cell. Although we did not get as far as a proof of the Jordan-Hölder and Krohn-Rhodes theorems, we can give some more insights into what it means for a semigroup to contain a group (see Fig. 23).

Definition 11. If S is a semigroup, the element $s \in S$ is **idempotent** iff $s^2 = s$.

The concept of idempotent can be seen as a generalisation for semigroups of the identity element for groups, although semigroups can have, independently, also an identity element. To see why, and to gain a better understanding of the significance of idempotents to our discussion, we provide an intuitive proof to the following assertion.

Assertion 12. Let S be a finite semigroup. For any $s \in S$, there exists an $n > 0$ such that s^n is idempotent.

Proof. The reader should refer to Fig. 23 for visual help on the discussion. To show that the assertion is true we need to find an integer n such that $s^{2n} = s^n$. Given any element $s \in S$, since S is finite, composing this element with itself indefinitely will eventually bring it back to a power k previously reached. In the case of groups $k = 1$, i.e. the element cycles back to itself (right after reaching the identity); in the case of semigroups, on the other hand, k can be greater than 1. Once the sequence of powers reaches s^k , the higher powers will equal elements already visited, thereby forming a closed cycle. Such a cycle is actually a cyclic group.

More precisely, let the number of elements forming this cycle be l . Consider the set of powers of s , $\{s^r : r \geq 1\}$; this is finite as S is finite, therefore there exist smallest k and l such that $s^k = s^{k+l}$. Then, \exists a smallest number m such that $n = ml \geq k$. In particular, if $l \geq k$, then $n = l$. That this is the case is fairly self-evident from Fig. 23. In many cases $l = k = 1$, which corresponds to a trivial cyclic group that has only one element, meaning that s is already an idempotent and all its powers are simply equal to itself.

That the element s^n acts as an identity for the cyclic group is easy to see. Let the cyclic group be G and let $s^n, s^r \in G$. Then, $s^r s^n = s^{r+n} = s^{r+ml} = s^r$. In fact, since l is the size of the group, adding ml to r simply cycles s^r m times around the loop, bringing it back to itself. Since $n \geq k$, this implies that, in particular (taking $r = n$), $s^n s^n = s^n$; hence s^n is idempotent. ■

A given cyclic group G embedded in a semigroup S can have more than one ‘tail’, meaning that it can be generated by more than one element of S . If $s, t \in S$ are two such elements and if n_1 and n_2 are their respective idempotent powers, then we must have that $s^{n_1} = t^{n_2}$ since the identity of a group G is unique. Such a case corresponds to more than one trajectory of states leading to the same set being permuted by G , as shown in Fig. 23.

Any kind of group, not just cyclic groups, can be found in a semigroup, of course. In particular, as already discussed in [5], the simple non-abelian groups (SNAGs) are very interesting from the point of

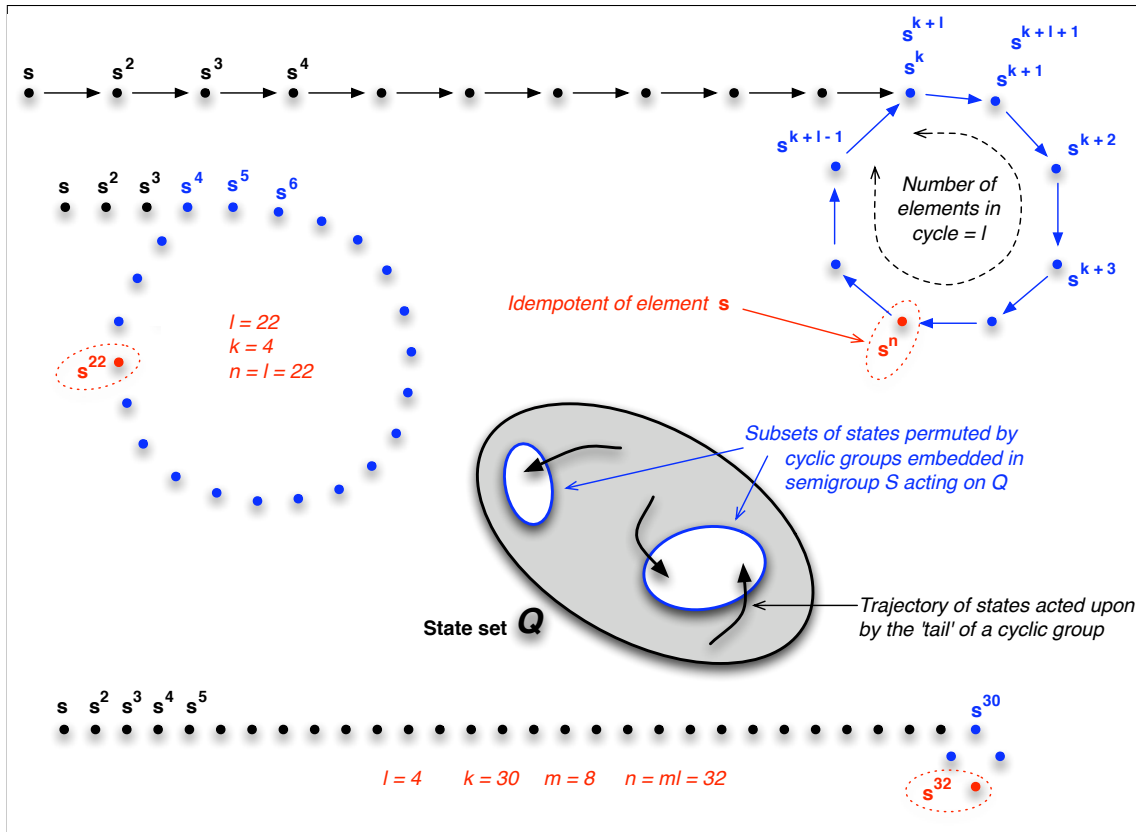


Fig. 23. Graphical summary of how cyclic groups can be embedded in a finite semigroup S , showing the idempotent for different cases, and how the subsets of states that correspond to the cyclic groups play the role of attractors within the state set Q of the automaton upon which S acts.

view of functional completeness and alternatives to Boolean algebra as a basis for computation [13]. The detailed discussion of the tetrahedron that we have developed across several project reports is relevant to the more general semigroup case because a semigroup analogue of Cayley’s theorem exists. What remains to be investigated is to what extent the generalisation of Cayley’s theorem to the cosets of a subgroup of a given group can be transported to a semigroup, where the action of its elements on a set of state can define any computable algorithm. It is not clear to what extent such a generalisation would be readily usable. A conservative view would be to say, at the very least, that in some cases the isomorphism between action on states and functional composition of the instruction set may lead to novel computational models or behaviour, but that in other cases no algebraic structure at all will be available. Being able to discriminate between the two, and understanding the computational properties of the algebraic case, would add a lot to our current understanding of the mathematics of computer science.

In the meantime, from a physical point of view the cyclic groups are interesting because they remind us of the limit cycles of dynamical systems, which can model the steady-state cyclic behaviour of driven, dissipative, open systems such as many cellular pathways.¹⁷ Clearly, we have only begun to map algebraic structures to computational structures, but we have hopefully been able to communicate how fundamentally interesting we find this approach at bio-inspired computing. In the final section of this appendix we turn now to the formal connections between behaviour and its realisation, through category theory.

¹⁷ In dynamical systems ‘steady state’ does not necessarily mean ‘not changing’. It generally refers to the *periodic* behaviour of systems driven by a periodic forcing function, which is to be contrasted with their transient behaviour. Transient behaviour is associated with the ‘free response’, meaning without driver, and is relevant just after the system has started from given non-equilibrium initial conditions or when the parameters of the driving function change during a run. If friction is present in the system, i.e. if it is dissipative, then the transient behaviour dies out over time and what’s left is the steady-state response, which is essentially the result of the *rate* of energy input from the driver having found a balance or equilibrium with the *rate* of energy dissipation through friction.

5 Categories

“[Category Theory] offers notions and a common language to describe structures independently of their internal complexity” ([59]: 30)

5.1 Introduction to Category Theory

We begin with a short introduction to category theory. We will explain the most important concepts relevant for our further research and to understand the following sections. After understanding the nature of category theory it will be easy to see how the studies in this domain connect to the investigation of the algebraic automata theory concepts presented in the previous section. Due to the nature of this section we used some textbooks and publications [85,86,58,51] relevant for discussing basic category.

To understand this subsection it is important to keep the general notion of category theory in mind. Basically, it is a highly abstract way to look at mathematics. It arose out of the need to have a formalism able to describe general characteristics of similar structures and to formalise transformations from one type of mathematical structure to another. We usually know mathematics from a rather *structure-internal* point of view, i.e. we look at a set of elements and define the operations we can perform on them to obtain the same or another set of elements. Category theory takes one step back and looks on the collection of these definitions from an *external* perspective. Thus, it does not discuss the elements of the internal structure but the structure described by the elements and their definitions and compares it with other structures. Using special types of functions it can also put these structures into relation or transform one structure into the other.

To get a better understanding of this concept we give a small example. Assume an injective set function f with $f : A \rightarrow B$. The characteristic of a function to be injective, or one-one, can be defined using the elements of the domain it is applied to. Thus, we could define that f is injective iff, for $x, y \in A$ and

$$f(x) = f(y), \quad \text{then} \quad x = y. \quad (28)$$

Now, assume that we have two parallel functions $g, h : C \rightarrow A$ with $f \circ g = f \circ h$. If we now take $x \in C$ we have $f \circ g(x) = f \circ h(x)$ which is equivalent to $f(g(x)) = f(h(x))$. As f is injective this equation is equivalent to $g(x) = h(x)$. Thus, it was very easy to show that if f is injective it has the property of being **left-cancellable**, which means that whenever

$$f \circ g = f \circ h \quad \text{then} \quad g = h. \quad (29)$$

If we invert this argument and assign f the characteristic of being left-cancellable we can show the following. Let's choose arbitrary $x, y \in A$ with $f(x) = f(y)$. Further, define a pair of *parallel functions* for which $g(0) = x$ and $h(0) = y$ without loss of generality. With these definitions, $f \circ g = f \circ h$. As f is left-cancellable, $g = h$ and thus $g(0) = h(0)$, which implies that $x = y$. Thus, if f has the left-cancellation property, it must be injective.

But why did we go through this example? Did we only want to learn that all injective set functions have the left-cancellation property? Our main intention behind this description was to show that we can totally abstract from the elements the functions above are acting on. Defining left-cancellation for a generic set function is independent of the elements the function is acting upon. We simply know that elements are picked from a certain set A , the domain, and mapped into another set B , the codomain.

How can we use this abstraction? To answer this question we first change the representation and illustrate our example in Fig. 24(a). This graph shows the sets A, B, C from our example as vertices. The functions are connecting these vertices by directed arrows. Let us now start with this abstract representation and translate it back into another domain. As an example, vertices may represent groups instead of sets and arrows may represent homomorphisms instead of functions. We then recognise that if we can define the same property of left-cancellation for groups as if the homomorphism between groups were injective, i.e. a monomorphism, then we can observe the same property.

In the same way, we can replace vertices, henceforth called **objects**, by other mathematical structures such as natural numbers, monoids, manifolds etc. Accordingly, edges, henceforth denoted as **arrows**,

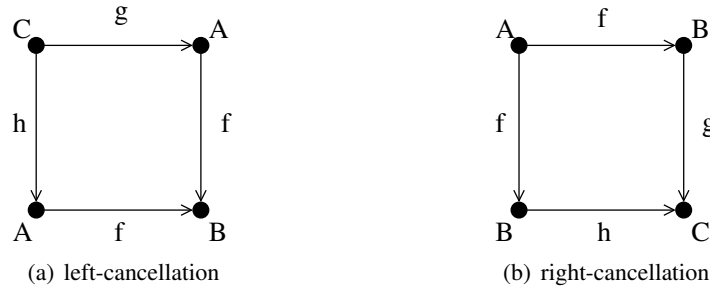


Fig. 24. Visualisation of left- and right-cancellation

are structure-preserving transformations that correspond to their respective structures, i.e. arrows may represent functions on natural numbers, set functions, homomorphisms, etc. Thus, independently of the internal structure of the elements we are investigating, e.g. no matter what kind of group we are dealing with, we can observe external structural characteristics.

This is on a very simple level the first important step towards understanding **categories**. They underpin the mathematical formalism that allows us to investigate these external structural characteristics, i.e. the combination of objects and arrows.

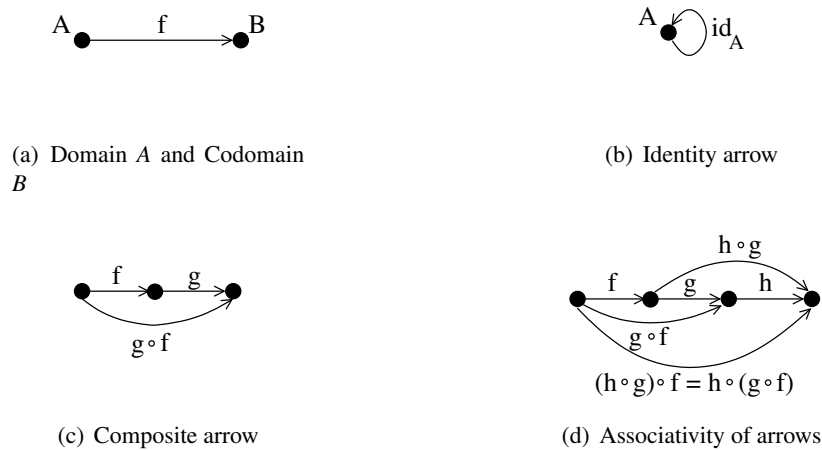


Fig. 25. Graphical representation of the category definition

Informally, a category C is defined by a class of objects and its corresponding arrows, also called morphisms. For each arrow f one object is defined as the **domain** of f (also denoted by $dom(f)$) and one object as the **codomain** also denoted by $cod(f)$ (see Figure 25(a)). For each object A in a category, there is an **identity morphism**. A is both the domain and codomain of this morphism. This identity morphism is denoted as id_A in Fig. 25(b). For each pair f, g of arrows in C with $cod(f) = dom(g)$ in C , i.e. $f : A \rightarrow B$ and $g : B \rightarrow C$, there is a **composite morphism** $g \circ f : A \rightarrow C$ (see Fig. 25(c)). Finally, in order to be a proper category **identity composition** and **associativity** have to hold. For identity composition, each arrow $f : A \rightarrow B$ in C has to comply with the equations

$$f \circ id_A = f \quad \text{and} \quad id_B \circ f = f \tag{30}$$

As expected, for associativity and the arrows $f : A \rightarrow B$, $g : B \rightarrow C$, and $h : C \rightarrow D$, the equation

$$(h \circ g) \circ f = h \circ (g \circ f) \tag{31}$$

has to hold (also expressed in the diagram in Fig. 25(d)). As usual the braces in this equation express precedence.

To emphasise the power of category theory, we can further abstract this concept and can also consider each partially ordered set as an object and monotone functions as the arrows between these objects. We obtain a category \mathcal{P} of all partially ordered sets.

Although we see that category theory is able to define very complicated structures in a very simple way, we take a step back and consider a very simple category, the category \mathcal{S} of sets. This will help us to easily understand basic but important characteristics of category theory. As already mentioned above the objects of \mathcal{S} are arbitrary sets. Its arrows are functions between sets.

We started this section with a characteristic called left-cancellation of an arrow (also see Eq. (29)). We can abstract this characteristic from \mathcal{S} – always an important step in category theory – and make a general statement for an arrow in any category \mathcal{C} . We call an arrow $f : A \rightarrow B$ in \mathcal{C} **monic** if for any pair of arrows $g, h : C \rightarrow A$ the equality $f \circ g = f \circ h$ implies that $g = h$. Functions or arrows in \mathcal{S} are exactly all injective homomorphisms, i.e. **monomorphisms**.

If we interpret the edges A, B , and C in Fig. 24(a) as objects of a category \mathcal{C} we can immediately see this property. In category theory, when showing or even proving special characteristics, it is common practice to draw a **diagram in** a category \mathcal{C} . We write **in** as the graphs shown in Figure 25 can be used to represent the whole structure **of** a category, i.e. they show the identity functions, their associativity, composition, etc. but do not pick some elements of the category and show some particular property of its construction. Thus, Fig. 24(a) shows us only the left- or right-cancellation property *within* a category but it does not describe the whole category. In contrast to this simple property description, Fig. 25(b) may already represent the description *of* a category consisting only of the object A and its identity arrow.

But why are diagrams so important in category theory? Many properties can be expressed by saying that a diagram **commutes**. A diagram commutes, if you can pick any two vertices in the graph and the arrows of the paths between these vertices determine new arrows which are all equal in the category \mathcal{C} you discuss. Thus, if the diagram in Fig. 24(a) commutes, this also implies Eq. (29). Thus, we could rephrase the definition of monic by saying: If the diagram in Fig. 24(a) commutes for any arrow g, h in \mathcal{C} then f is monic.

If on the contrary the reverse diagram (as shown in Fig. 24(b)) commutes, we obtain a new characteristic of the arrow f , namely

$$g \circ f = h \circ f \quad \text{then} \quad g = h, \quad (32)$$

which is also called right-cancellable or **epic**. In \mathcal{S} epic arrows are surjective. Accordingly, we call surjective homomorphisms **epimorphisms**.

We just implicitly discovered a new notion in category theory: **duality**. By simply exchanging domain and codomain and reversing composition a new characteristic is obtained. This can be done for any statement σ in a category \mathcal{C} . Its **dual** or **opposite** statement is denoted by Σ^{op} . In fact, the same can be done for a given category \mathcal{C} : all objects in \mathcal{C} become objects of \mathcal{C}^{op} and for any arrow $f : A \rightarrow B$ in \mathcal{C} we add an arrow $f^{op} : B \rightarrow A$ in \mathcal{C}^{op} . Of course, composition in \mathcal{C}^{op} is inverted too, i.e. for any $f \circ g$ in \mathcal{C} we introduce $g^{op} \circ f^{op}$ in \mathcal{C}^{op} . Below, We will see more of these examples.

After finding an equivalent concept for injective and surjective set functions, the next obvious question is: Is there also a similar concept for bijective set functions in category theory? There is such an arrow characteristic. It is called **iso** but is defined slightly differently, which also implies that from an arrow which is *monic* and *epic* does not directly follow that this arrow is also *iso*. In category theory an arrow $f : A \rightarrow B$ is *iso* in \mathcal{C} – also referred to as invertible – if there is a \mathcal{C} arrow $g : B \rightarrow A$ such that $g \circ f = id_A$ and $f \circ g = id_B$. It can be shown that there is at most one such g .

So why is the notion of iso different from the notion of a bijective function? If we take our simple set category \mathcal{S} we observe that every arrow which is monic and epic is also iso. However, consider the category induced by a partially ordered set (P, \leq) . If an arrow $f : A \rightarrow B$ in this set has an inverse $f^{-1} : B \rightarrow A$, then $A \leq B$ and $B \leq A$ which means that $A = B$. This implies that f must be the unique arrow id_A . As a consequence, in a category induced by a partially ordered set all arrows are monic and epic but the only iso's are the identities. An isomorphism $f : A \rightarrow B$ directly induces the **isomorphic objects** A and B , which is denoted by $A \cong B$.

Another important set of objects in category theory are **initial** and **terminal** objects. From an initial object $\mathbf{0}$ in a category \mathcal{C} there is exactly one single arrow to any other object A in \mathcal{C} . By construction, if

there is more than one initial object, these objects are isomorphic in \mathcal{C} . Categories with one initial object are the category of all sets and the category induced by a poset. If we now apply the concept of duality to the definition of an initial object we obtain the definition of the terminal object of a category. Thus, an object $\mathbf{1}$ in a category \mathcal{C} is terminal if there is exactly one arrow for each element A in \mathcal{C} from A to $\mathbf{1}$. As initial objects, all terminal objects are isomorphic too. Typical examples for the initial objects in a category are the empty set in the category of sets and the minimum or bottom element in a partial order. Equivalently, all singletons in the category of sets and the maximum or top element in the category of a partial order represent terminal objects.

So far we have defined quite generic characteristics of arrows and objects. They do not reveal the full power of category theory but find a rather simple analogy in all categories. However, we are now going to define an important operation which is often complicated to define for different types of objects: the **product**.

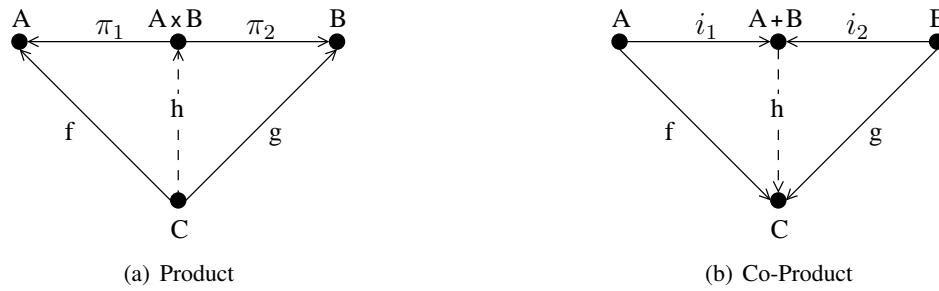


Fig. 26. Product and Co-Product representation

Defining products is usually domain-specific. We basically use set theory to define the product for sets, groups, topological spaces, etc. Each of them is defined separately and differently but carries the same name, i.e. it is a product. Category theory has the power to identify the common characteristics of these different products and gives a general definition of a product: For two objects A and B the product $A \times B$ is an object such that there are two morphisms

$$\pi_1 : A \times B \rightarrow A \quad \text{and} \quad \pi_2 : A \times B \rightarrow B. \quad (33)$$

Further, for any two morphisms f, g in \mathcal{C} with

$$f : C \rightarrow A \quad \text{and} \quad g : C \rightarrow B \quad (34)$$

there is a unique morphism $h : C \rightarrow A \times B$ such that the diagram in Fig. 26(a) commutes.

As the concept of commutative diagrams may be pretty new to the reader we derive this characteristic as an example. Assume that we define the product set and use the following notation

$$A \times B = \langle x, y \rangle : x \in A \text{ and } y \in B \quad (35)$$

Further, we may define h, π_1 and π_2 by the rules $h(x) = \langle f(x), g(x) \rangle$, $\pi_1(\langle f(x), g(x) \rangle) = f(x)$, and $\pi_2(\langle f(x), g(x) \rangle) = g(x)$. π_1 and π_2 are also called **projections** as they take an element of the product set and map it into the corresponding set which helps to form the product set. We can directly see that by these rules the equations $\pi_1 \circ h = f$ and $\pi_2 \circ h = g$ hold. We have defined f, g , and h in such a way that the diagram in Fig. 26(a) commutes. Thus, h , which can also be denoted as $\langle f, g \rangle$, is exactly the product map as defined in \mathcal{S} . Up to isomorphism this product as well as any other product in a category is unique.

Let's consider a more interesting product: the product in a partial order (P, \leq) . If this product of the two elements p, q exists it is defined by the following properties:

$$p \times q \leq p \quad \text{and} \quad p \times q \leq q \quad (36)$$

and

$$\text{if } c \leq p \text{ and } c \leq q, \quad \text{then } c \leq p \times q. \quad (37)$$

Both equations are a direct consequence of the definition of the product and the fact that in the category induced by this partial order the arrows are equivalent to the relations in the partial order. Further, we can observe that Eq. (36) is equivalent of defining a *lower bound* on p and q . Eq. (37) refines this definition and defines the product as a *greatest lower bound* in the partial order (P, \leq) .

By duality the product directly induces the definition of the so-called **co-product** (see Fig. 26(b)). Here, we exchange the names of the original projections and replace them by so-called **injections** i_1 and i_2 . They are defined by $i_1 : A \rightarrow A + B$ and $i_2 : B \rightarrow A + B$. Thus, if we denote $h(x)$ by $[f, g]$, called the co-product arrow, we can also write $[f, g] \circ i_1 = f$ and $[f, g] \circ i_2 = g$. Again, this is a very generic and general definition. As above, we can specify this definition and derive the co-product for sets, which is equivalent to the disjoint union of sets A and B . This in fact is also the reason the co-product is often called the **sum**.

As expected, in a partial order (P, \leq) , the co-product of two elements of P represents the least upper bound of the two elements. This can be shown in the similar fashion as done above for the greatest lower bound. We know from [1] that a partial order, in this case with a least upper bound and a greatest lower bound, represents a lattice. In categorical terms we can redefine a lattice: a skeletal preorder with a product and co-product for each two elements is a lattice.

If we now take a step back and look at Fig. 26 we should be able to recognise that we constructed a very special object. Why is this? In order to define our product, for example, we had to define the arrows in such a way that the product is the domain of the arrows whose codomains are A and B . We also mentioned that it can be shown that this element is unique up to isomorphism. Thus, the product has a **universal property** amongst all other elements which come into question. The construction which yields this special element is also often called **universal construction**.

But why are universal properties and their construction so important for category theory? If it is possible to define or find universal properties within a category or even between them, some proofs or investigations often become easier, more efficient and concise. Investigating and understanding the rather abstract properties often help to understand problems or other aspects about the same property just in a slightly different situation. Remember that we partially showed above how a product for all types of set categories can be defined. Investigating this product once enables us to find out specific things for products in sets, partial orders, groups, etc.; it enables us to understand the construction for all of them and the relations between them.

It is possible to define the universal property in a much more formal way, which makes use of unique morphisms and commuting diagrams. We abstain from presenting this formal definition here, in the hope that we were able to communicate the gist of universal properties. We continue with a basic but very important concept in category theory: **functors**.

So far, this section has dealt with arrows which connect objects within a category, preserving the structure of these objects. Functors abstract even further and act on a higher hierarchical level: instead of transforming objects, they transform whole categories while preserving their internal structure. They allow the transformation, connection, description and investigation of different types of categories. Thus, they are basically part of the essence of category theory. In fact, S. Eilenberg and S. MacLane consider "... the whole concept of a category ... [as] ... essentially an auxiliary one". For them the "... the basic concepts are essentially those of a functor and of a natural transformation".

As functors are such an important concept in category theory and because we are going to make use of this concept in this document and in future work, we give a semi-formal definition for a functor. A functor F from a category \mathcal{C} to a category \mathcal{D} is a function which transforms

- each object c in \mathcal{C} into an object $d = F(c)$ in category \mathcal{D}
- and each arrow $f : A \rightarrow B$ in \mathcal{C} into an arrow $g = F(f)$ in category \mathcal{D} with $F(f) : F(A) \rightarrow F(B)$,

with the following restrictions

$$F(id_A) = id_{F(A)} \quad (38)$$

$$F(g \circ f) = F(g) \circ F(f) \quad \text{if } g \circ f \text{ is an arrow in } \mathcal{C}. \quad (39)$$

These restrictions preserve the identities (see Eq. (38)) in the new category and ensure that commutative diagrams in \mathcal{C} remain commutative in \mathcal{D} . Obviously, a functor preserves the basic internal structure of a category, i.e. the direction of arrows, identities, commutativity. Such functors are called **covariant functors**. As an example you may imagine a functor which maps a category \mathcal{C} into a **subcategory** of \mathcal{D} , i.e. \mathcal{D} is a “bigger” category which contains \mathcal{C} .

Due to the principle of duality we also expect the dual of covariant functors. They exist and are called **contravariant functors**. The only difference to covariant functors is that in the contravariant case all arrows in the category are reversed, i.e. we find the same identities and the same – if not removed by merging some objects – commutative diagrams with the main difference that the arrows are reversed.

Before we continue and link the very theoretical category theory to the more practical algebraic automata theory we need to explain one final concept also important for the subsequent argumentation.

Of course, with the powerful concept of functors we can not only map categories from one category to another one. In fact, functors are comparable to arrows or even more abstract, they can be objects. For now, we want to stick with the notion of a functor as an arrow. We have seen how we can define monic and epic arrows and how they induce iso arrows. The same can be done to be able to compare categories. A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ is said to be iso if it has an inverse functor G with $G \circ F = id_{\mathcal{C}}$. Thus, can we use an iso functor to say that two categories are isomorphic, or equivalent? Yes, we can. However, does this make sense or is this condition too restrictive? If we take a closer look at the definition of a functor we will find a direct answer. Yes, it is too restrictive! A functor directly maps an object A into its equivalent $F(A)$. Thus, if we apply our inverse functor G we obtain equality in $A = G(F(A))$, which is far stronger than an isomorphism where only $A \cong G(F(A))$ is required.

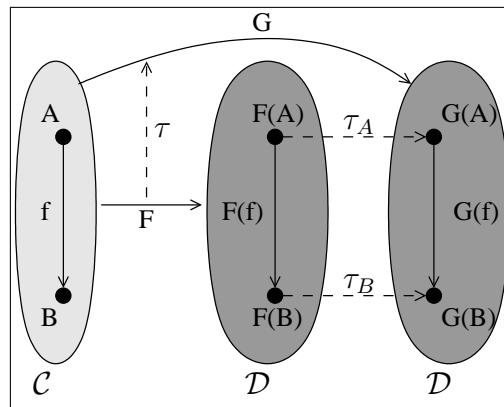


Fig. 27. Natural transformation of functor F into functor G

Let’s assume we have already found a functor F which is an isomorphism. How do we adapt it in order not to be as restrictive as before? This is done by applying the concept of a **natural transformation** τ which allows us to transform functor $F : \mathcal{C} \rightarrow \mathcal{D}$ into another functor $G : \mathcal{C} \rightarrow \mathcal{D}$, i.e. $\tau : F \rightarrow G$. For each element A in \mathcal{C} , τ provides a so-called **component** τ_A which is a simple assignment $\tau_A : F(A) \rightarrow G(A)$ such that for any arrow $f : A \rightarrow B$ in \mathcal{C} it is

$$\tau_B \circ F(f) = G(f) \circ \tau_A \quad (40)$$

If every component of τ is iso, i.e. there exists an inverse $\tau_A^{-1} : G(A) \rightarrow F(A)$ for every $\tau_A : F(A) \rightarrow G(A)$, we call τ a **natural isomorphism** and denote it by $\tau : F \cong G$. Please note that this isomorphism has nothing to do with isomorphic categories. Instead this isomorphism defines the notion of isomorphic functors. In fact, natural transformations allow the definition of a new type of category in which functors are the objects of the category and the natural transformations represent the arrows of this category.

We can now use the natural isomorphism and the categories they induce to define the **equivalence of categories**: this equivalence corresponds to a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ for which an inverse functor $G : \mathcal{D} \rightarrow \mathcal{C}$ exists such that there are natural isomorphisms $\tau : id_{\mathcal{C}} \cong G \circ F$ and $\delta : id_{\mathcal{D}} \cong F \circ G$.

5.2 Connecting Automata Theory and Specification

Impact of Category Theory on Automata Theory As we said above, all types of algebraic structures can form categories. Category theory offers appropriate tools to work on these structures. It can put different categories into relation with each other or even map them into each other. Thus, one obvious question that we need to answer is whether we can also apply category theory to automata or transition systems in general, i.e. whether we could use the powerful tools of category theory to compare the results we get from discussing systems or models of real systems in biology, physics, etc. with systems or models in automata theory.

Can we directly map automata or even any kind of system into category theory? Partially, this question has already been answered in the last section and in Section 4. We mapped the tetrahedron into a permutation group, which is automatically also a monoid. From the last section we also know that objects of a category can be monoids. Thus, the simplest category representing the tetrahedron DFA is a category consisting of one object, the set of all elements of the monoid. The arrows of this category are the transitions of the automaton. This mapping is very easy. However, it does not reveal much about the internal structure of the automaton. Thus, we choose a more structured mapping according to [87], which better distinguishes between the single elements the automaton is working on, its inputs and the transformations it experiences.

Assume an automaton definition with the signature: $\mathcal{A} = (A, Q, \lambda)$. Further, let G be the semigroup of automaton \mathcal{A} constructed from the input alphabet A with the operation of concatenation. From this semigroup it is straightforward to construct a special category: a **small category**. A small category simply consists of objects which are all sets but not classes, such as groups, monoids, etc. Here it also becomes obvious why a monoid is a special class of category. If we simply map each element of semigroup G into an object and all transformations between two elements in G into arrows between the corresponding objects we obtain a new small category \mathcal{G} .

So far only the *input processing* to our automaton has been considered. Thus, we already know a lot about *what* the automaton actually processes. But, we still have to clarify the internal structure of our automaton, i.e. *how* it processes the input. For this purpose we consider the category \mathcal{S} of all sets. We further define a functor with domain \mathcal{G} and codomain \mathcal{S} . It maps each object $u \in \mathcal{G}$ to set S of the automaton. With each arrow $z \in \mathcal{G}$ we associate the corresponding mapping $\lambda^*(z)$ defined by the original signature of the automaton. Thus the functor enables us to separate the internal structure of the automaton – the state transitions – from the set of objects it is working on, e.g. the language, i.e. the set of words, the automaton accepts or processes.

Obviously the whole construction depends on the semigroup G , the set of states to be reached depending on the input words to the automaton, and the system of transitions between the states, which is the description of the automaton. In this construction, the functor describes the mappings between a set of ‘words’ processed in a certain state of the automaton and the state transitions between these sets. Thus, the functor which performs the mapping between the category of a semigroup and the category of sets can be interpreted as an automaton. Hence, the categorical and thus functorial view allows us “...to distinguish the automaton’s structure and those mathematical objects on which this structure is realised” [87].

As outlined in the last section, we can further interpret each of these functors as an object and construct a more abstract category $\mathcal{S}^{\mathcal{G}}$. Morphisms between objects of this category, i.e. automata, would then be natural transformations. They allow us to compare different automata, construct new automata by combining functors, discuss their similarities or differences etc.

This algebraic discussion of automata in the previous sections finds its complement in the **coalgebraic** point of view. Coalgebra exactly exploits the notion of duality presented above and discusses automata from a different point of view. Very briefly, we will give these two different types of interpretation a more formal and categorical background and show why category theory offers a framework ideal for studying automata in an algebraic domain.

As we have seen, we can describe an automaton starting from its internal structure as presented above. Its definition is based on a **carrier** set Q which is also called the set of states. Together with a transition function $\lambda : Q \times A \rightarrow Q$ it describes the behaviour of the automaton. With the knowledge and

terminology from above we can form a so-called F -algebra. It is described by the tuple (Q, α_Q) with an endofunctor $F : \mathcal{S} \rightarrow \mathcal{S}$ and the morphism $\alpha_Q : F(Q) \rightarrow Q$. This is a very general description which allows us – depending on the type of functor F – to describe many different classes of automata. As an example, we can use functor $F(M) = A \times M$ to describe an automaton with a fixed input alphabet A . Together with F the morphism α_Q describes how to *construct* new elements in Q .

If we now apply the principle of duality we obtain a different notion of interpretation, an F -coalgebra [88]. Analogously to F -algebras there is an endofunctor $F : \mathcal{S} \rightarrow \mathcal{S}$ and a morphism $\beta_Q : Q \rightarrow F(Q)$ which is basically the dual to α_Q (we find an inverted arrow). Similarly, an F -coalgebra is defined by the tuple (Q, β_Q) and is similarly able to describe an automaton. However, coalgebras have a different kind of expressiveness about the characteristics of the elements of Q . In contrast to the algebraic point of view, we don't have full details about how to construct the elements. Instead the coalgebraic approach simply describes which operations to apply in order to obtain a more *complete* view on the set Q . One typical example for a coalgebra is the specification of a recursive data type such as a list.

More precisely, let's consider again the F -coalgebra from above. To be able to define the system (Q, β_Q) entirely, we will need two injections (see above) $i_{\beta_Q} : Q \rightarrow A$ and $k_{\beta_Q} : Q \rightarrow Q$ in order to define $\beta_Q : Q \rightarrow T(Q)$. If we now want to know more about Q and A we will need to consider one single state $q \in Q$. With this state, we are able to compute $x = i_{\beta_Q}(q) \in A$ and the next state of our automaton. In so doing subsequently, we obtain a sequence of states and the *word* which belongs to this state sequence by simulation. If two such sequences are similar for two different starting states $q_1, q_2 \in A$ then these states are called bisimilar. Accordingly, if there are two such automata defined over the same functor F the process of comparing the two automata yields the so-called process of bisimulation in the domain of coalgebras which is basically dual to finding congruent algebras in the domain of algebras [89].

At this point one may wonder why these different notions of algebraic presentation of automata have been investigated, why we are presenting them here, and why we are dealing with highly abstract category theory. The answer is relatively simple and we have answered it implicitly before. Most of the observations already made in the well-studied domain of algebras already hold in the domain of coalgebras by the principle of duality, e.g. the observations on quotients and subsystems. Thus many results can be transferred into another domain. On the other hand, each domain provides special insights. Often some properties are easier or better to describe in one domain compared to the other. Similarly, some solutions to particular problems become easier to solve in the dual domain. E.g. in the domain of algebras the finding of subalgebras is related to identifying initial algebras which are minimal. In coalgebras the “dual notion” is the identification of final or terminal coalgebras which have no proper quotients and thus are simple. Additionally, F -coalgebras and F -algebras which are based on one specific functor form special types of categories which are also called topoi [86]. Observations in this category can help to get a deeper understanding of the class of automaton one is dealing with, e.g. finding isomorphisms to other automata or sub-automata, etc. In the end, it turns out that the combination of *universal algebra* and *universal coalgebra* using the “glue” category theory is a powerful tool to formally describe a large variety static and dynamic systems in computer science, starting from programming language semantics, finite and infinite or recursive data types, transition systems, concurrent programming languages, dynamical systems, etc.

Among the various results which have been presented in pertinent literature we would like to mention the results from Gabriel Ciobanu and Sergiu Rudeanu [90], which nicely emphasise the latter points. Using category theory they were able to translate existing links between Mealy, Moore, and Rabio-Scott automata into isomorphisms of categories, thus not only showing equivalences between single automata but between different classes of automata.

Jacobs [91] combines non-deterministic and probabilistic systems which have been studied and understood separately. For technical and conceptual reasons the combination of both system domains was difficult. However, based on the work of Varacca [92] who combined both domains, coalgebra offers a way to describe traces in such systems, which is also important for validation and verification purposes. Badouel and Tchendji [93] use principles of coalgebras to ensure consistent update mechanisms of hierarchically organised documents whose parts – considered as subsystems – are fragmented in highly distributed computer networks.

This section showed how algebra and coalgebra can be used to investigate and specify a large variety of systems with the help of category theory. Whereas the area of algebraic concepts is rather well investigated and a mature area, the research domain of coalgebra is just conquering the various fields of its possible application. One of these fields is coalgebraic logic and coalgebraic automata specification, which we will briefly discuss in the following section.

Automata Specification and Logic Algebraic specification finds its roots in the wish to define a formal and thus mathematical model in order to be able to verify certain properties of a system. It can be helpful to specify simple structures, e.g. when defining abstract data types for the interfaces of sub-systems a complex system consists of. But algebraic specification is not only perfectly suitable for these smaller systems, such as safety-relevant components in cars or airplanes, but it is also appropriate to consistently design larger system, such as complex software systems.

In the previous sections we showed how it is possible to completely describe a system by simply using an algebra, consisting of a carrier set and some operations working on this set, an algebra. Thus, the same algebra or a subset of it can be used to specify the system, e.g. in terms of what the system *must* or *must not* do, e.g. because some security policy prohibits the transition into a specific state. With the help of category theory we are able to refine, combine, or analyse these specifications. However, the previous section also showed us that category theory defines the dual concept of algebra, coalgebra. Hence, there is not only the area of algebraic specification but also of coalgebraic specification. An obvious question suggests itself: Where is the difference? We already know that both theories are complementary and that it is sometimes better to use one or the other. The same holds true for (co-)algebraic specification. But, when do we use which? This answer is not easy as it certainly depends on the specific characteristic to be specified and the system this specification is determined for. However, a general guideline is already suggested by the definition of the respective theories. We know that in the case of an algebra we have complete knowledge on the state space. Thus, in case of algebraic specifications we can choose a specification ‘language’ which uses the state space itself such as complete traces. In the case of a coalgebraic specification, on the other hand, the internal state of the system is not important as we have only limited knowledge of the state space. Instead, the operations on the state space which transition the system into new states are of importance and thus are intrinsically used for specification purposes.

Of course, the type of specification does not only depend on the property to be specified but also on the type of system it is determined for. It turns out that also the logic used to specify a particular system property strongly depends on the system description (mainly due to Stone Duality [94]). In [1] we have listed a small choice of logics and how their expressiveness can be used to specify a system. They are perfectly feasible to specify properties in classical systems. Of course, it is desirable to be able to also use temporal logic for reasoning about systems specified by the coalgebraic approach. Also for the purpose of specifying requirements or constraints some type of logic is essential. Hence, we want to sketch how the concepts of temporal logic also apply to the domain of (co-)algebraic specifications.

Coalgebraic logic was (probably) first defined by Moos [95]. He defines general modal logics interpreted on coalgebraic systems. This approach allows the definition of any type of logic by specifying some modality. As mentioned above, we are going to use the modality of time to describe how linear temporal logic – at least a subset of it – can be tightly linked to coalgebra and thus how it can be used for verification purposes.

First of all, we need to specify predicates. As indicated before, we can express certain characteristics of a system by selecting some states and giving them a predicate, e.g. “terminal”, “not allowed”, etc. Thus, predicates in our domain are sets P, R of the state space Q (recall that in our example we defined the automaton $\mathcal{A} = (A, Q, \lambda)$), i.e. $P, R \subseteq Q$. The following point-wise definition of the Boolean connectives is along the same lines as presented in [96]:

$$\neg P = \{q \in Q \mid \neg P(q)\} \quad (41)$$

$$P \wedge R = \{q \in Q \mid P(q) \wedge R(q)\} \quad (42)$$

$$P \vee R = \{q \in Q \mid P(q) \vee R(q)\} \text{ etc.} \quad (43)$$

In this way it is relatively straightforward to define the simple Boolean connectives. The same holds true for the temporal operators. Let $\alpha_Q : Q \rightarrow F(Q)$ be a coalgebra defined as above. Further, let $Pred$ be a function for which

$$\forall q \in P \Rightarrow \alpha_Q(q) \in Pred(F)(P). \quad (44)$$

$Pred$ is basically an invariant which holds true for the whole system. With this predicate we can easily define the *next* operator $\bigcirc : \mathcal{P}(Q) \rightarrow \mathcal{P}(Q)$ by:

$$\bigcirc P = \alpha_Q^{-1}(Pred(F)(P)) \quad (45)$$

$$= \{q \in Q \mid \alpha_Q(q) \in Pred(F)(P)\}. \quad (46)$$

As $Pred(F)(P)$ expresses an invariant of the system which holds true, we can interpret the next operator as the set of states which are direct successors of state q and fulfil the predicate P . Thus, with a very simple specification, we are able to express the same meaning as explained in [1] using graphs. For an in-depth discussion of this type of mapping of temporal logic into coalgebraic logic we also recommend [97] and [98].

Conclusion: Where we are going This section gave just a very small insight to complex category theory, tried to outline how algebras and coalgebras can be put in relation, and gave some short overview how the latter systems relate to automata theory and their logic counterparts.

The various and security-relevant applications of category theory and (co)algebra make us feel confident about the direction of our research. Currently we are striving towards a deeper comprehension of how we can directly couple automata specifications and their logical equivalent. Our understanding of the rather ‘young’ bialgebraic approach of using a combination of an algebraic and coalgebraic specification to describe automata on the structural and functional level, respectively, is a first step in this direction.

More important are the recent findings in the field of trace semantics [99,91] which yield a description of traces in the framework of coalgebra. Why is this? Traces are known to be an important means to identify security properties in systems. When used for specifying security properties, i.e. setting a security policy for a system, particular traces can even be combined in order to express more complex security policies. Of course, in the framework of coalgebra traces are defined using a coalgebra. Thus, it will be interesting to study the impact of the framework of coalgebra on systems using traces to specify security properties, e.g. as presented by Mantel in [100] to specify information flow properties. We think that these traces may yield a particular category, probably a subcategory of the more generic traces presented by Hasuo et al. [99]. Its functorial influence on different types of algebras may yield new results on the construction and design of automata. What does this mean?

Assume that we can develop a logical specification system in which the system description has direct impact on the semantics and structure of an automaton. Further assume that we can show that this coupling of specifications and their respective automata induces a special class of category such as is done in classifying categories. In this case we might be able to represent automata with special characteristics, such as certain security properties, as a subcategory of the category of all possible automata the interaction computing framework defines. This will not only restrict the type of automata to particular structures but it may also intrinsically limit the number and type of combinations with other automata. Current research – to our knowledge – has not been able to show yet whether the theory of F -coalgebras can be developed in its corresponding functor F . This observation however would greatly help to develop the research outlined above. Fortunately, active research in the area of modal logic and the efforts of trying to discover a direct relationship between the coalgebraic logics and their respective systems represents a viable alternative.

5.3 Categorical Foundations for Symbiotic Security

In this next section we cover more advanced concepts of category theory, which will lead to a formal presentation of the relationship between automata structure and behaviour through adjunctions.

In interaction computing the traditional approach at designing or specifying a system must be changed. This is due to the fact that the approach to realise a machine and its computation in interaction computing is fundamentally different from the traditional von-Neumann architecture. Interaction computing needs to support a high level of concurrency, interdependability, non-determinism, probability, etc., to name only a few properties of the framework requirements. In fact, this list is potentially not complete. From the various deliverables it is clear that the analytical language we use is algebra and category theory and its powerful tools developed over the last decades. This chapter describes how the combination of both domains can help define powerful concepts which help to link a particular machine to a category of machines, and how this category is related to a category of behaviour this machine realises.

We have already observed that the interaction computing approach, which we can basically associate with the workings of biochemical systems and cellular processes, is far too complex to be described through its single details. Thus, we have to use feasible mechanisms which, on the one hand, abstract from the complex details implementing a certain behaviour but, on the other hand, allow a concise and precise description and translation of the interaction machine to be implemented. Thus, we challenge the currently established methodology of system specification by applying a purely demand-driven approach enabled by concepts such as symbiotic computing, which is derived from interaction computing. Instead of focusing on implementing details, e.g. on security on a software or hardware level, we focus on the application-level semantics. As a consequence, the means to describe machines in the interaction computing framework must be a strictly behaviour-based specification language.

The goal of this language is to abstract from the realisation details in terms of architecture and implementation of the machine which is supposed to implement the specified behaviour and to give the *designer* of a system the possibility to specify mainly behavioural aspects of the machine to be generated. A translation process transparently generates the complete machine in a rigorous manner using operational semantics, whose definition will form one of the next areas of activity in our research. This will not be ‘yet another specification language’, since it is inspired by the links between observable biological symmetries and their corresponding behaviour. This chapter investigates an important method in category theory, which helps achieve such a mapping.

This mapping will establish new connections in the field of algebra, coalgebra, and as a consequence in category theory. It will rely on work which links structure – represented by algebra – with its corresponding behaviour – represented by coalgebra. In particular, so-called behaviour-realisation adjunctions will be investigated, which were first recognised by Goguen in 1972 [101,53,54]. They form the basis for deriving an appropriate operational semantics. Similar work conducted in 1981 by G.D. Plotkin [102], which layers coalgebra on top of algebra, will support this task. The more specialised framework of universal (co)algebra, an important branch of category theory, will be relevant for this research. However, it is not within the focus of this chapter. In fact, together with the previous section, the material discussed and refined in this deliverable forms the foundation for understanding universal (co)algebra. While the translation process from algebras into logic [103,25] has been understood fairly well, universal algebra and algebraic logic will complement this understanding and offer tools for comparing and studying logics. This will help us cover new ground in the fields of coalgebraic logic and universal coalgebra and identify the required elements for the specification language described above.

In order to remain manageable and scalable, the specification language will also have to rely on language features which allow to build hierarchies, known from the paradigm of object-oriented programming. Thus, the design of the language will also rely on existing research, e.g. J.A. Goguen and G. Malcolm’s 2000 [57], which was also based on the foundations presented in this chapter.

We strive for the definition of a synthesis process which is guided by specifying application-relevant, user-centric behaviour. This will finally enable us to redefine symbiosis from a security point of view. Security is usually bound to a particular application, i.e. security services usually require some application or service to run which, in turn, requires some security service to be executed. This establishes a symbiotic relationship between the service to be protected and its corresponding security service. As a consequence, by design, our approach cannot yield interaction machines which do not comply with some specified security characteristic or which execute unnecessary security services. This would not only introduce a completely new thinking in the security domain but would also – if not specified otherwise –

keep security transparent to the user and/or designer. Furthermore, security can be seen as a perfect *application* for exploring the power of symbiotic computing to implement security services and for developing new methods to remedy the currently established practice of security patching. Thus, along these lines this section complements the foundations provided in the previous ones, to develop a real alternative to the security mechanisms that secure the deployment of automatically composed or modified services in a semi-automated ex-post patching manner.

Simple Functorial Operations on Sets From the very beginning of our discussions on abstract algebras [1], we assumed familiarity with basic concepts such as basic operations on sets. This section is going to refresh the definitions of some of these operations and explain how they fit into the framework of categories and why they have functorial character. The understanding of this characteristic will be important for the next section in which these operations will be combined with polynomial functors.

Product One of the most common operations on sets is the Cartesian product. For simplicity we recapitulate the binary Cartesian product. Formally, it is defined as the set

$$X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}, \quad (47)$$

where X and Y are two arbitrary sets.

From the previous section we recall the notion of a projection π . It takes an element of the product set and maps it into one of the sets which was used to build the product. More specifically, we talk about $\pi_1 : X \times Y \rightarrow X$ and $\pi_2 : X \times Y \rightarrow Y$, defined by $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$.

These projections become quite useful if we consider the product not only on sets but also on functions. For this purpose we construct a pairing function $\langle f, g \rangle : Z \rightarrow X \times Y$, where $f : Z \rightarrow X$ and $g : Z \rightarrow Y$. $\langle f, g \rangle$ can then be defined in a straightforward fashion by $\langle f, g \rangle(z) = (f(z), g(z)) \in X \times Y$ with $z \in Z$.

With this construction it becomes obvious that we can apply our projection functions not only on sets but also on functions. Explicitly, we can apply π_1 and π_2 to our pairing function and obtain

$$\pi_1 \circ \langle f, g \rangle = f \quad (48)$$

$$\pi_2 \circ \langle f, g \rangle = g. \quad (49)$$

Thus, the given projection functions can also be applied to our composed function $\langle f, g \rangle$. So, if we collect the facts presented above, the product allows the combination of two functions, whereas the projections π_1 and π_2 represent the inverse operation.

This is generally identical to what the product acting on sets does. Thus, we can define the abstract product $f \times g : X \times Y \rightarrow X' \times Y'$ with

$$(f \times g)(x, y) \mapsto (f(x), g(y)). \quad (50)$$

This construction is equivalent to

$$f \times g = \langle f \circ \pi_1, g \circ \pi_2 \rangle. \quad (51)$$

Further, we recognise that

$$\langle \pi_1, \pi_2 \rangle = id_{X \times Y}, \quad (52)$$

which implies that

$$\begin{aligned} \langle id_X, id_Y \rangle &= \langle id_X \circ \pi_1, id_Y \circ \pi_2 \rangle \\ &= id_{X \times Y}. \end{aligned} \quad (53)$$

Further, let $h : W \rightarrow Z$ and $k : W \rightarrow Z$ be two functions. Then we can construct two functions $f \circ h : W \rightarrow X$ and $g \circ k : W \rightarrow Y$ for which we obtain:

$$\begin{aligned} (f \circ h) \times (g \circ k)(w) &= ((f \circ h)(w), (g \circ k)(w)) \\ &= (f(h(w)), g(k(w))) \\ &= (f \times g) \circ (h(w), k(w)) \\ &= (f \times g) \circ (h \times k)(w). \end{aligned}$$

In short, the equation

$$(f \circ h) \times (g \circ k) = (f \times g) \circ (h \times k) \quad (54)$$

states that the product applied to functions also preserves composition.

With this knowledge in mind we look back to the information provided in the previous subsection. We defined a category \mathcal{C} as

1. a collection of objects,
2. a collection of arrows,
3. operations which assign to each arrow f its domain and codomain,
4. a composition operator which assigns to each pair of arrows f and g with their appropriate domains and codomains its composition $f \circ g$, and
5. an identity arrow for each object in \mathcal{C} .

Further, we defined a functor F as a transformation between two categories \mathcal{C} and \mathcal{D} . The latter do not have to be different or distinct. The functor F has to satisfy the following characteristics and properties:

1. each object $c \in \mathcal{C}$ is transformed into an object $d \in \mathcal{D}$ by $d = F(c)$,
2. each arrow $f \in \mathcal{C}$ with $f : A \rightarrow B$ is transformed into an arrow $g \in \mathcal{D}$ with $g = F(f)$ and $F(f) : F(A) \rightarrow F(B)$,
3. each identity arrow is transformed in an identity arrow of the new category, i.e. $F(id_A) = id_{F(A)}$, and
4. each composition $f \circ g \in \mathcal{C}$ is transformed into its appropriate composition $F(f \circ g) = F(f) \circ F(g)$, i.e. the functor preserves composition.

This explicitly shows that the product can be applied to the category \mathcal{S} . The objects of \mathcal{S} are sets and the total functions between them are the arrows of this category. Thus, the product, as defined here, has not only functorial properties as they apply to sets *and* functions but it is in fact a functor which maps the product category $\mathcal{S} \times \mathcal{S}$ to the category \mathcal{S} .

Coproduct While discussing duality, the coproduct $+$ was defined by inverting projections pi_1 and pi_2 to injections ι_1 and ι_2 . They are defined as $\iota_1 : X \rightarrow X + Y$ and $\iota_2 : Y \rightarrow X + Y$. As this type of function is hard to grasp we introduce a general example of the set $X + Y$:

$$X + Y = \{(x, 1) | x \in X \cup (y, 2) | y \in Y\}.$$

Here we can see why the coproduct is also often called disjoint union. It *labels* each element of the corresponding sets with 1 and 2 respectively. Thus, within the coproduct we can still distinguish the elements from each set the coproduct was generated from. As a consequence we can define the injection functions, also called *coprojections*, as $\iota_1 : X \rightarrow X + Y$ and $\iota_2 : Y \rightarrow X + Y$ with $\iota_1(x) = (x, 1)$ and $\iota_2(y) = (y, 2)$.

Similarly, for the (inverted) functions $f : X \rightarrow Z$ and $g : Y \rightarrow Z$ we can compose an appropriate coproduct function $[f, g] : X + Y \rightarrow Z$. It is defined by

$$[f, g](w) = \begin{cases} f(x), & \text{if } w = (x, 1) \\ g(y), & \text{if } w = (y, 2) \end{cases} \quad (55)$$

We now extend the coproduct concept to functions by first listing the following characteristics we can observe with coproducts and their injection functions:

$$\begin{aligned} [f, k] \circ \iota_1 &= f \\ [f, k] \circ \iota_2 &= g \\ [\iota_1, \iota_2] &= id_{X+Y} \end{aligned} \quad (56)$$

With these observations and the definition of $[f, g]$ we can construct a coproduct which can also be applied to functions. Let $d : X \rightarrow X'$ and $e : Y \rightarrow Y'$. The coproduct on functions $d + e : X + Y \rightarrow X' + Y'$ can then be defined by

$$(d + e)(w) = \begin{cases} (d(x), 1), & \text{if } w = (x, 1) \\ (e(y), 2), & \text{if } w = (y, 2). \end{cases} \quad (57)$$

With the injection functions ι_1 and ι_2 , we can rewrite this definition to $d + e = [\iota_1 \circ d, \iota_2 \circ e]$. With h, k we can observe the following characteristic of this construction

$$\begin{aligned} ((d \circ h) + (g \circ k))(w) &= \begin{cases} ((f \circ h)(x), 1), & \text{if } w = (x, 1) \\ ((g \circ k)(y), 2), & \text{if } w = (y, 2) \end{cases} \\ &= \begin{cases} (f(h(x)), 1), & \text{if } w = (x, 1) \\ (g(k(y)), 2), & \text{if } w = (y, 2) \end{cases} \\ &= (f + g) \circ \begin{cases} (h(x), 1), & \text{if } w = (x, 1) \\ (k(y), 2), & \text{if } w = (y, 2) \end{cases} \\ &= ((f + g) \circ (h + k))(w) \end{aligned} \quad (58)$$

Additionally, we find that

$$(id_X + id_Y)(w) = \begin{cases} (id_X(x), 1), & \text{if } w = (x, 1) \\ (id_Y(y), 2), & \text{if } w = (y, 2) \end{cases} \quad (59)$$

$$= id_{X+Y}(w). \quad (60)$$

Again, we see that the product has functorial properties and in fact satisfies the conditions of a functor. The coproduct transforms the product category of sets $\mathcal{S} \times \mathcal{S}$ into the category of sets \mathcal{S} . In Section 5.3 we will see why these characteristics are important. Before showing this, we introduce two more operations which also have functorial properties: powerset and exponent.

Powerset Another well-known operation on sets is the powerset $\mathcal{P}(X)$ which constructs the set consisting of all subsets of X , i.e. $\mathcal{P}(X) = \{U \mid U \subseteq X\}$. While the definition on sets is straightforward, we can show that the powerset operation can also be applied to functions. For this purpose we define $f : X \rightarrow Y$. Further, we define the operation $\mathcal{P}(f)(U) : \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ on f as follows:

$$\begin{aligned} \mathcal{P}(f)(U) &= \{f(x) \mid x \in U\} \\ &= \{y \in Y \mid \exists x \in X, f(x) = y \vee x \in U \wedge x \in U\} \end{aligned}$$

Again, this construction yields a functor $\mathcal{P}(-) : \mathcal{S} \rightarrow \mathcal{S}$. This is due to the fact that for the identity it obviously holds that

$$\begin{aligned} \mathcal{P}(id_X)(U) &= \{id_X(x) \mid x \in U\} \\ &= \{x \mid x \in U\} = U \\ &= id_{\mathcal{P}(-)}, \end{aligned}$$

where $U \subseteq X$. Additionally, the powerset operation also preserves composition due to the following equation system

$$\begin{aligned} \mathcal{P}(f \circ g)(U) &= \{f \circ g(y) \mid y \in U\} \\ &= \{f(x) \mid x \in \{g(y) \mid y \in U\}\} \\ &= \{f(x) \mid x \in \mathcal{P}(g)(U)\} \\ &= \mathcal{P}(f)(\mathcal{P}(g)(U)) \\ &= (\mathcal{P}(f) \circ \mathcal{P}(g))(U), \end{aligned} \quad (61)$$

where $g : Y \rightarrow Z$ and $U \subseteq Y$.

Exponentiation Finally, to be able to construct polynomial functors we consider the last operation: exponentiation. Given two sets X and Y one can define the set of all total functions, i.e. functions which are defined for each element in X . Thus, $Y^X = \{f \mid f \text{ is a total function } X \rightarrow Y\}$ which is called the exponent of X and Y . Similarly to the projection and injection functions defined for product and coproducts, respectively, exponentiation defines the evaluation function $eval : (Y^X \times X) \rightarrow Y$ and the abstraction function $\Lambda(g) : Z \rightarrow Y^X$ for a function $g : Z \times X \rightarrow Y$. On input (f, x) , with $f : X \rightarrow Y$ and $x \in X$, the evaluation function $eval$ yields as output $f(x) \in Y$, i.e.

$$eval(f, x) = f(x). \quad (62)$$

Any particular argument $z \in Z$ to the abstraction function $\Lambda(g)$, on the other hand, determines a function Y^X . Thus, if g_z represents a particular *curried* version of g , then $g_z(x) = g(z, x)$. This implies $\Lambda(g)(z) = g_z$. While it is fairly easy to show that

$$eval \circ (\Lambda(g) \times id_X) = g, \quad (63)$$

$$\Lambda(eval) = id_{Y^X}, \text{ and} \quad (64)$$

$$\Lambda(h \times id_X) = h, \quad (65)$$

it is rather complicated to understand that exponentiation can also have functorial characteristics and that we can even construct a exponentiation functor.

Let us first recollect and refine the definition of a contravariant functor we gave above. So far, we have seen covariant functors which simply map objects to objects and arrows to arrows preserving their *direction*. Contravariant functors also map objects to objects but they apply duality and reverse the direction of the arrows. Thus, a functor $F : \mathcal{C}^{op} \rightarrow \mathcal{D}$ is called contravariant on \mathcal{C} . More explicitly: such a functor F would transform an arrow $e : U \rightarrow V$, with U and V as objects of \mathcal{C} , and transform it into $F(e) : F(V) \rightarrow F(U)$ and $F(d \circ e) = F(e) \circ F(d)$, where d is also an arrow in \mathcal{C} .

So, why do we discuss contravariant functors? It is required because we will define an exponent functor $\mathcal{S}^{op} \times \mathcal{S} \rightarrow \mathcal{S}$ which is obviously contravariant in the first argument.

In order to introduce functorial properties of our exponentiation we need to define a functor which does not only map ordinary sets or – more generally – objects, to their corresponding function space but also all arrows between these sets, which are themselves functions (at least if defined on sets). Thus, we need a function which is able to transform a function space Y^X into another one, say V^U . To do this, we define two functions, $k : X \rightarrow U$ in \mathcal{S}^{op} , and $h : Y \rightarrow V$ in \mathcal{S} . We do not need to define k in \mathcal{S}^{op} and thus make the functor invariant because there are other constructions for the exponent functor. However, this construction is more intuitive. This is due to the fact that we can now simply apply $h^k : Y^X \rightarrow V^U$ on a function $f \in Y^X$ defined by

$$h^k(f) = h \circ f \circ k \in V^U. \quad (66)$$

At first sight this may appear wrong as $k : X \rightarrow U$. However, in this equation k is a function $k : U \rightarrow X$ defined in \mathcal{S} and according to the explanation about the application of a contravariant functor to functions, this equation is correct. It maps the function space Y^X to V^U as desired.

It remains to show that h^k preserves identity and composition. For identity, the equation

$$(id^{id})(f) = id \circ f \circ id = f \quad (67)$$

holds. For composition we choose $h_1, h_2 : Y \rightarrow V$ defined in \mathcal{S} and $k_1, k_2 : X \rightarrow U$ defined in \mathcal{S}^{op} to show that

$$\begin{aligned} (h_2^{k_2} \circ h_1^{k_1})(f) &= (h_2^{k_2}(h_1 \circ f \circ k_1)) \\ &= h_2 \circ h_1 \circ f \circ k_1 \circ k_2 \\ &= ((h_2 \circ h_1)^{(k_2 \circ k_1)})(f). \end{aligned} \quad (68)$$

Cartesian Closed Categories The mapping properties defined above, i.e. product, co-product, powerset, and exponentiation, are ‘universal mapping properties’ (UMP) and can be used to form a particular notion which will be helpful for our future work as it has great practical relevance: Cartesian Closed Category (CCC). Together with a terminal object, the binary product and exponentiation on all pairs of objects in category \mathcal{C} form a CCC.

As we just went through the definition of the various UMPs in \mathcal{S} listed above we can already state that, together with a terminal object,¹⁸ \mathcal{S} forms a Cartesian closed category. Cartesian closed categories are important for our research as a tight correspondence with typed λ -calculus can be identified. This correspondence has been pointed out in papers by Lambek and Scott [104]. Due to space limitations of this article and due to the exploratory character of this work we can currently only sketch the relationship between these two concepts. Nevertheless, we give a short introduction to the abstract programming language of typed λ -calculus and show how it corresponds to a Cartesian closed category.

The typed λ -calculus consists of a basic set T of base types. They include the terminal type, denoted by *unit*, the product type, denoted by $A \times B$ for each pair $A, B \in T$, and a functional type, denoted by $A \rightarrow B$, again defined over each pair $A, B \in T$. With these *ingredients*, the observing reader may already guess where the correspondence may arise. To actually see it we need to first look at the syntax and semantics of the typed λ -calculus.

According to [52] we can define an abstract and simple grammar for an expression M in λ -calculus:

$$M ::= \text{unit} \mid c \mid x \mid \lambda x.M \mid (M M') \mid (M, M') \mid \text{fst } M \mid \text{snd } M$$

The grammar contains a metavariable c with type B_c . c reflects an element from the set of constants C . The same applies to the variable x . $\lambda x.M$ reflects a functional abstraction, in which we may find free variables, i.e. all variables in expression M which are not x . An expression (MM') denotes a function application and (M, M') a pairing with the corresponding projection function *fst* and *snd* which map to the first and second argument, respectively.

Obviously, with these components, we can easily define a category with binary products for a fixed λ -calculus \mathcal{L} . The associated category of types $\mathcal{C}(\mathcal{L})$ can be defined by the types as objects, arrows by equivalence classes of closed terms (terms with no free variables). The identities of this category are $1_{B_x} = \lambda x.x$ where x has type B_x . Composition of arrows c and d corresponds to $\lambda x.c(bx)$.

With the definition for exponentials given in Section 5.3 it is sufficient to show that Equations 63 and 65 are satisfied in order to show that $\mathcal{C}_{\mathcal{L}}$ is Cartesian closed. This is due to the fact that the category above is already well defined and that it already contains a binary product. Thus, we choose for any pair of objects A, B the evaluation function $\text{eval} : B^A \times A \rightarrow B$ defined by

$$\text{eval} = \lambda z.f\text{st}(z)\text{snd}(z)$$

where z has type B_z . For any arrow $f : Z \times A \rightarrow B$ we define the abstraction Λ by

$$\Lambda = \lambda z \lambda x.f\langle z, x \rangle : Z \rightarrow B^A$$

where z and x are of type B_Z and B_A respectively. Using strict lambda calculus we can now show that Equations 63 and 65 hold.

Functors and their (Co)algebras We start this section with a very simple and intuitive example. In Section 4, we discussed many different notions of groups. In general, we can say that a group is based on a set together with some binary operation which combines two elements to construct a new element which is again part of this group. Of course, this *structure* has to satisfy additional constraints to become a group but for now we ignore these requirements. Instead we simply look at the given *structure*, i.e. the operations on the underlying set, say G . In Fig. 28 we use categorical means to represent this structure. Reading this structure with category theory it is possible to apply the knowledge learnt above, in particu-

¹⁸ The terminal objects in \mathcal{S} are all the singletons of \mathcal{S} , i.e. all the 1-element sets.

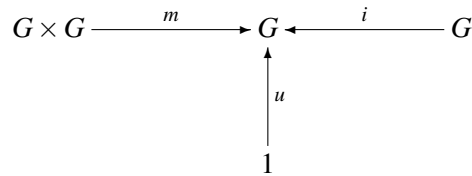


Fig. 28. Abstract group structure ($m = \text{closure}$, $i = \text{inverse}$, $u = \text{identity}$)

lar the knowledge about the simple operations product, coproduct, etc., and can construct a single arrow as follows

$$1 + G + G \times G \xrightarrow{u,i,m} G$$

which basically induces a functor $F : \mathcal{S} \rightarrow \mathcal{S}$ defined by

$$F(X) = 1 + X + X \times X.$$

Thus, the structure of a group can be represented by a single arrow. It is actually possible to generalise this idea and define so-called *F-algebras*.

F-Algebras *F*-algebras are given by an endofunctor $F : \mathcal{C} \rightarrow \mathcal{C}$ where \mathcal{C} is an arbitrary category. The *F*-algebra consists of an object $C \in \mathcal{C}$ and an arrow

$$\gamma : F(C) \rightarrow C.$$

As for regular groups which can be transformed by a homomorphism we also define homomorphisms of *F*-algebras on elements in C , i.e. we define $h : (C, \gamma) \rightarrow (C', \gamma')$ such that arrow $h : C \rightarrow C'$ in \mathcal{C} satisfies the equation $h \circ \gamma = \gamma' \circ F(h)$. We obtain a category of *F*-algebras: $F - Alg(\mathcal{C})$.

What does this mean? We just defined a means to describe the algebraic structure of simple algebraic constructs such as groups, rings, etc. They are called simple, because the number of finite operations must also be finite. Thus, a functor of the form

$$F(X) = C_0 + C_1 \times X + C_2 \times X^2 + \dots + C_n \times X^n$$

is called a *polynomial functor* and can represent sanitary structures. Of course, in the general case C_i has to be finitely and the number n of operations needs also to be finite. We can also call this functor a type of *signature* as it basically lists all operations allowed in the algebra described by F . The object C is called the carrier of the *F*-algebra.

As an example we consider the zero, $\mathbf{0} : 1 \rightarrow \mathbb{N}$, and successor functions, $s : \mathbb{N} \rightarrow \mathbb{N}$, defined on the natural numbers, \mathbb{N} . They form an algebra $[\mathbf{0}, s] : 1 + \mathbb{N} \rightarrow \mathbb{N}$ of the functor $F(X) = 1 + x$. Please note that we explicitly write that $[\mathbf{0}, s]$ is an algebra of F . We could also write that $[\mathbf{0}, s]$ is one of the algebras of F . Even more precise would be to say that $[\mathbf{0}, s]$ is a model of F . Therefore, we consider the functor as a signature of operations. The signature of the algebra consists of the different operations plus the carrier set. Thus, our example algebra above defines the signature $(\mathbb{N}, [\mathbf{0}, s])$.

As we know that we can find many different algebras of the same signature, i.e. for the same functor, it is obvious to ask whether there is some means to transform these different algebras into one another while preserving their characteristics (signature). We call this a “homomorphism of algebras”. Given two *F*-algebras (U, a) and (V, b) defined through the same functor F this homomorphism consists of a function $f : U \rightarrow V$ which maps the carrier sets to one-another, and which commutes with the operations $f \circ a = b \circ T(f)$.

This definition becomes clearer if we look at the two algebras defined by a and b . They are defined by

$$\begin{aligned}
 a &: F(U) \rightarrow U \\
 b &: F(V) \rightarrow V
 \end{aligned}$$

$$\begin{array}{ccc}
 F(U) & \xrightarrow{F(f)} & F(V) \\
 \downarrow a & & \downarrow b \\
 U & \xrightarrow{f} & V
 \end{array}$$

Fig. 29. Algebra homomorphism f between F -algebras (U, a) and (V, b)

With this knowledge we can draw the commuting diagram shown in Fig. 29.

Adding another F -algebra, e.g. (W, c) , to our discussion, we can observe that the resulting algebra homomorphisms $f : U \rightarrow V$ and $g : V \rightarrow W$ are composable to an algebra morphism $g \circ f : U \rightarrow W$. With the help of Fig. 30 this can be shown by the following equalities

$$\begin{aligned}
 g \circ f \circ a &= g \circ b \circ F(f) \\
 &= c \circ F(g) \circ F(f) \\
 &= c \circ F(g \circ f),
 \end{aligned}$$

which identify $(g \circ f)$ as an algebra homomorphism by definition.

$$\begin{array}{ccccc}
 F(U) & \xrightarrow{F(f)} & F(V) & \xrightarrow{F(g)} & F(W) \\
 \downarrow a & & \downarrow b & & \downarrow c \\
 U & \xrightarrow{f} & V & \xrightarrow{g} & W
 \end{array}$$

Fig. 30. Composition of algebra homomorphisms f and g between F -algebras (U, a) and (V, b) and (V, b) and (W, c) , respectively.

In Section 5.1 we introduced initial and terminal objects of categories. An initial object was up to isomorphism basically defined as an object which has exactly one arrow to each other object in the category. With the algebra homomorphism we can introduce the interesting concept of initiality for algebras. We consider an algebra $i : F(U) \rightarrow U$ of functor F and carrier set U to be *initial* if for each other algebra $a : F(V) \rightarrow V$ there is a unique algebra homomorphism from (U, i) to (V, a) . It can be shown (see also [88]) that if such algebra exists then it is unique, up-to-isomorphism, and that it also has an inverse $i^{-1} : U \rightarrow F(U)$.

F-Coalgebras Similarly to F -algebras we can define F -coalgebra. Obviously, the only difference for an F -coalgebra is the structure signature, i.e. of functor F . Instead of applying F to the carrier set it maps C to $F(C)$.

$$\gamma : C \rightarrow F(C).$$

Equivalently to F -algebras we can also define homomorphisms of coalgebras from one F -coalgebra (U, a) to another coalgebra (V, b) defined over the same functor F . It is also represented by a function $f : U \rightarrow V$ between the carrier sets. However, for coalgebras f has to satisfy $a \circ f = F(f) \circ b$ with $a : U \rightarrow F(U)$ and $b : V \rightarrow F(V)$. The small but important difference to the homomorphism of algebras becomes clearer when looking at the categorical representation of f in Fig. 31. As a consequence of dealing with coalgebras we can also define the dual concept of initiality for F -coalgebras, i.e. finality. We consider a coalgebra $e : Z \rightarrow F(Z)$ as final if for every coalgebra $c : U \rightarrow F(U)$ there is a unique map of coalgebras $(U, c) \rightarrow (Z, e)$.

At this point the reader should have an abstract understanding of F -algebras and their initiality and F -coalgebras and their finality. We take one step back and want to emphasise the difference between algebras and coalgebras using a more informal and general representation.

$$\begin{array}{ccc}
 U & \xrightarrow{f} & V \\
 a \downarrow & & \downarrow b \\
 F(U) & \xrightarrow{F(f)} & F(V)
 \end{array}$$

Fig. 31. Coalgebra homomorphism f between F -coalgebras (U, a) and (V, b)

In Section 5.2 we have already indicated that coalgebras take a different point of view when describing or specifying automata. We explained that while the nature of algebra is more structural the nature of coalgebra is more dynamical. In contrast to an algebra the coalgebra does not specify or construct the whole set of states it is actually working on. The latter is basically the principle of algebra which is also called induction. From one single element it is possible to derive the whole state space; thus, all elements are basically described from the very beginning. In coalgebra this is different as the single elements are derived by recursively applying a particular function on, e.g. an input or previous state. This function is some kind of black box and it becomes only clear which kind of state transition takes place after actually observing the output of this black box.

With this observation in mind we have a look at final algebras. We can define various coalgebras for a particular functor. All of them can be mapped to the final coalgebra which is unique up to isomorphism. This entails that the final coalgebra contains all possible variations of coalgebras including all of their potential *behaviour* (below we specify what we mean by behaviour). Thus, the final coalgebra is some kind of minimal representation for the possible coalgebras which can be described by F . On the other hand, the initial algebra is basically the maximum degeneration of all coalgebras which are models of F .

Adjunctions One very important concept – some authors even consider it to be the most important concept [52,105,58] – of category theory are adjoints or adjunctions. They have not been discussed yet as they require some deeper understanding of category theory as many concepts of category theory are and can be applied, in particular the universal mapping properties.

By formalising adjunctions in 1958, Daniel Kan offered a framework which allowed the rather simple description and connection of universal properties. Their expressiveness and thus importance is reflected by the *wonders of adjunctions* David E. Rydeheard lists in [55]:

- The widespread occurrence of adjunctions in mathematics and programming.
- That trivial functors determine, through adjunctions, constructions of great interest in applications.
- Adjunctions often provide abstract descriptions of the sort of symbolic tasks with which programming is concerned.

This list should already give the reader a hint why we are actually investigating adjunctions. Their relevance for our research should become clear in the remainder of this chapter.

So what are adjunctions? Unfortunately, the concept of adjoints is rather intricate. Therefore, we start with giving a general idea of what an adjunction is, and then we give an example for a simple adjunction before finally formally defining what an adjoint actually is using categorical terms.

Definition by natural transformations As expected, adjunctions represent a particular relationship between two categories, say between \mathcal{C} and \mathcal{D} . This relationship is not as rigorous as equality. In contrast, it describes an even weaker – better: looser but richer – relationship than isomorphism. Assume that our two categories are related by the functors F and G in the following way

$$\mathcal{C} \xrightleftharpoons[G]{F} \mathcal{D}. \tag{69}$$

If \mathcal{C} and \mathcal{D} were isomorphic then

$$\begin{aligned}
 1_{\mathcal{C}} &= G \circ F \text{ and} \\
 F \circ G &= 1_{\mathcal{D}},
 \end{aligned}$$

where 1_C and 1_D represent the identities in categories C and D , respectively.

Adjunctions represent an even weaker concept than that, even weaker than an equivalence relation between these categories. An isomorphism is basically a special case of an equivalence. Instead of transforming one element in one category and back into the same category, an equivalence relation would transform these elements up to isomorphism. Thus, $F \circ G$ and $G \circ F$ would only transform an element of its respective category into another isomorphic one. This can be represented as

$$1_C \cong G \circ F \text{ and} \\ F \circ G \cong 1_D.$$

While equivalences already give a lot of freedom to the composition of $G \circ F$ and $F \circ G$, adjunctions finally implement an even weaker concept as it does not map elements up to isomorphism but it uses natural transformations (introduced in Section 5.1, but see also [106,58]). This can be represented by

$$1_C \xrightarrow{\bullet} G \circ F \text{ and} \\ F \circ G \xrightarrow{\bullet} 1_D.$$

Of course, we cannot simply take any natural transformation which fits this pattern, but need to set up some axioms which restrict the lenient notion of an adjunction to a useful level.

So with this very simple introduction we can already give a rather formal definition of adjunctions: Given two functors $F : C \rightarrow D$ and $G : D \rightarrow C$, an adjunction, denoted by $G \dashv F$, is defined by a pair of natural transformations,

$$\eta : 1_C \xrightarrow{\bullet} G \circ F \text{ and} \\ \varepsilon : F \circ G \xrightarrow{\bullet} 1_D,$$

The natural transformations cannot be chosen arbitrarily but they must ensure that the identity triangles in Fig. 32 commute. In the remainder of this section we call η the *unit* and ε the *co-unit*. We call them units

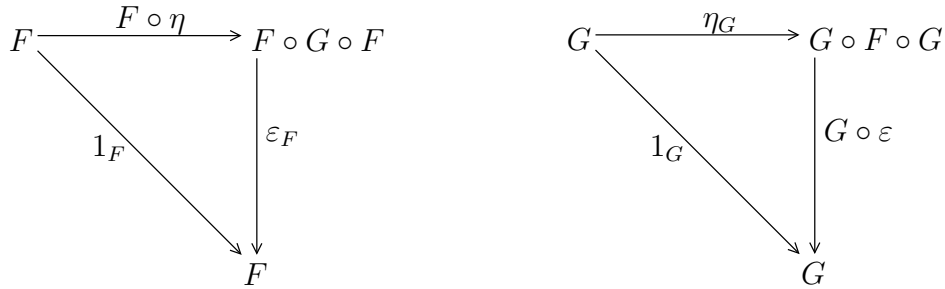


Fig. 32. Identity triangles in C and D respectively

because with the isomorphisms $\phi \in \text{Hom}_C(F(X), Y) \cong \text{Hom}_D(X, G(Y))$ and $\psi \in \text{Hom}_D(X, G(Y)) \cong \text{Hom}_C(F(X), Y)$ we can define

$$\eta_X = \phi(1_{F(X)}) \quad (70)$$

$$\varepsilon_Y = \psi(1_{G(Y)}) \quad (71)$$

So what do these natural transformations actually tell us about the arrows in the respective category? To better understand the concept of an adjunction we additionally give an equivalent definition which considers adjunctions from the point of view of the morphisms (arrows) within the categories.

Definition by isomorphic morphisms From this point of view and given the functors above, we define an adjunction $G \dashv F$ as a natural isomorphism in $X \in \mathcal{C}$ and $Y \in \mathcal{D}$ between the homomorphisms within the categories \mathcal{C} and \mathcal{D} respectively:

$$\text{Hom}_{\mathcal{D}}(F(X), Y) \cong \text{Hom}_{\mathcal{C}}(X, G(Y)),$$

What does this equation basically express? First, we observe that for these sets to be isomorphic there needs to be transformations for all morphisms $f \in \text{Hom}_{\mathcal{D}}(F(X), Y)$ which transform each f into its transposed counterpart \bar{f} . The same has to hold for each $g \in \text{Hom}_{\mathcal{C}}(X, G(Y))$.

So, we need to understand what the functors are which allow for this natural isomorphism. To show that the isomorphism is natural in X and Y we first fix Y . Thus, an object $X \in \mathcal{C}^{op}$ needs first to be mapped to $\text{Hom}_{\mathcal{D}}(F(X), Y)$. This is done by a functor $\mathcal{C}^{op} \xrightarrow{\mathcal{D}(F, Y)} \mathcal{S}$, where $_$ represents a placeholder for any object in \mathcal{C} . Similarly, a functor $\mathcal{C}^{op} \xrightarrow{\mathcal{C}(X, G, _)} \mathcal{S}$ maps X into $\text{Hom}_{\mathcal{C}}(X, G(Y))$. This can be better understood by drawing the naturality square, shown in Fig. 33. Now, the question is in which way are

$$\begin{array}{ccc} \text{Hom}_{\mathcal{D}}(F(X), Y) & \xrightarrow{\bar{_}} & \text{Hom}_{\mathcal{C}}(X, G(Y)) & \text{Hom}_{\mathcal{D}}(F(X), Y) & \xrightarrow{\bar{_}} & \text{Hom}_{\mathcal{C}}(X, G(Y)) \\ \downarrow _ \circ F \circ f & & \downarrow _ \circ f & \downarrow g \circ _ & & \downarrow G \circ g \circ _ \\ \text{Hom}_{\mathcal{D}}(F(X'), Y) & \xrightarrow{\bar{_}} & \text{Hom}_{\mathcal{C}}(X', G(Y)) & \text{Hom}_{\mathcal{D}}(F(X), Y') & \xrightarrow{\bar{_}} & \text{Hom}_{\mathcal{C}}(X, G(Y')) \\ \text{(a) in } X \text{ with } \forall f, f : X \rightarrow X' & & & \text{(b) in } Y \text{ with } \forall g, g : Y \rightarrow Y' & & \end{array}$$

Fig. 33. Naturality square for the natural isomorphism $\text{Hom}_{\mathcal{D}}(F(X), Y) \cong \text{Hom}_{\mathcal{C}}(X, G(Y))$

these definitions similar. We will show that by identifying unit and counit, i.e. defining the two natural transformations η and ε , as in the previous definition.

For this purpose, we put $Y = F(X)$, that is, we look at the identity morphism $1_{F(X)}$ of $F(X)$. As we have an isomorphism the identity $1_{F(X)}$ in \mathcal{D} has to be transformed into an identity in \mathcal{C} , i.e. in our case it has to be a morphism in $\text{Hom}_{\mathcal{C}}(X, G(F(X)))$. As you may remember, we called the natural transformation $1_{\mathcal{C}} \xrightarrow{\bullet} G \circ F$, above, η . Due to its similarity, we call this transformation η'_X , for now. Similarly, if we consider the case $X = G(Y)$, i.e. we look at the identity $1_{G(Y)}$, we can derive from Fig. 33 that $F \circ G$ should be equivalent to the identity $1_{\mathcal{D}}$ in \mathcal{D} . Again, we find an equivalent notion from the discussion before, $\varepsilon : 1_{\mathcal{D}} \xrightarrow{\bullet} F \circ G$. Therefore, we call this functor, similarly, ε'_Y .

With η' we can derive the following equations from Fig. 33(a):

$$\eta'_X \circ f = \overline{F \circ f}, \quad \text{for } Y = F(X) \text{ and} \quad (72)$$

$$\varepsilon'_Y \circ F \circ f = \bar{f}, \quad \text{for } X = G(Y). \quad (73)$$

This derivation is pretty straightforward. For example, consider Equation 72. We start at the top-left corner with an identity morphism $1_{F(X)} \in \text{Hom}_{\mathcal{D}}(F(X), F(X))$ and transform it into the appropriate identity morphism in $\text{Hom}_{\mathcal{D}}(F(X'), Y)$ by applying $_ \circ F \circ f$, which results in the functor $1_{F(X)} \circ F \circ f = F \circ f$. Finally, to arrive at $\text{Hom}_{\mathcal{C}}(X', G(Y))$, we have to transpose $F \circ f$. On the right side of the naturality square on X we already have the unit functor η' , which transforms X into $G(F(X))$. To arrive at $\text{Hom}_{\mathcal{C}}(X', G(Y))$ too, we simply have to transform X into X' before applying η' .

Similarly, we can derive the following pair of equations for the naturality in Y . According to Fig. 33(b) the following equations hold

$$G \circ g \circ \eta'_X = \bar{g}, \quad \text{for } Y = F(X) \text{ and} \quad (74)$$

$$g \circ \varepsilon'_Y = \overline{G \circ g}, \quad \text{for } X = G(Y). \quad (75)$$

In the last step, we finally need to verify the naturality of η' and ε' and show the satisfaction of the identity triangles as seen in the previous definition. As a result, we will obtain the equalities $\eta' = \eta$

$$\begin{array}{ccc}
 X & \xrightarrow{\eta'_X} & G(F(X)) \\
 \downarrow f & & \downarrow G \circ F \circ f \\
 X' & \xrightarrow{\eta'_{X'}} & G(F(X')) \\
 \text{(a) Square for } \eta' & &
 \end{array}
 \qquad
 \begin{array}{ccc}
 Y & \xleftarrow{\varepsilon'_Y} & F(G(Y)) \\
 \downarrow g & & \downarrow F \circ G \circ g \\
 Y' & \xleftarrow{\varepsilon'_{Y'}} & F(G(Y')) \\
 \text{(b) Square for } \varepsilon & &
 \end{array}$$

Fig. 34. Naturality squares for the isomorphism η' and ε'

and $\varepsilon' = \varepsilon$. To verify the naturality of η' we use Fig. 34. We can follow two different paths, namely: $X \xrightarrow{f} X' \xrightarrow{\eta'_{X'}} G(F(X'))$ and $X \xrightarrow{\eta'_X} G(F(X)) \xrightarrow{G \circ F \circ f} G(F(X'))$ to arrive at $G(F(X'))$ starting at X . Thus, in order for the naturality square to commute the following equality has to hold

$$\begin{aligned}
 \eta'_{X'} \circ f &= G \circ F \circ f \\
 \overline{F \circ f} &= G \circ F \circ f \circ \eta'_X && \text{(with Eq. 72)} \\
 \overline{F \circ f} &= G \circ (F \circ f) \circ \eta'_X \\
 \overline{F \circ f} &= \overline{F \circ f} && \text{(with Eq. 74)}
 \end{aligned}$$

Thus, we have shown the naturality of η' . In the analogous manner we can show the naturality of ε' . Again, there are the paths $F(G(Y)) \xrightarrow{\varepsilon'_Y} Y \xrightarrow{g} Y'$ and $F(G(Y)) \xrightarrow{F \circ G \circ g} F(G(Y')) \xrightarrow{\varepsilon'_{Y'}} Y'$ in 34(b). We can again verify that this square commutes by looking at the following equations

$$\begin{aligned}
 g \circ \varepsilon'_Y &= \varepsilon'_{Y'} \circ F \circ G \circ g \\
 \overline{G \circ g} &= \varepsilon'_{Y'} \circ F \circ G \circ g && \text{(with Eq. 75)} \\
 \overline{G \circ g} &= \varepsilon'_{Y'} \circ F \circ (G \circ g) \\
 \overline{G \circ g} &= \overline{G \circ g} && \text{(with Eq. 73)}
 \end{aligned}$$

It remains to show that the identity triangles for η' and ε' also commute. This time we construct the identity triangles in Fig. 35 and verify their commutation.

$$\begin{array}{ccc}
 F(X) & \xrightarrow{F \circ \eta'_{F(X)}} & F(G(F(X))) \\
 \searrow 1_{F(X)} & & \downarrow \varepsilon'_{F(X)} \\
 & & F(X)
 \end{array}
 \qquad
 \begin{array}{ccc}
 G(Y) & \xrightarrow{\eta'_{G(Y)}} & G(F(G(Y))) \\
 \searrow 1_{G(Y)} & & \downarrow G \circ \varepsilon'_{G(Y)} \\
 & & G(Y)
 \end{array}$$

Fig. 35. Identity triangles for η' and ε'

Example First let us recall from Section 5.1 the definition of a preorder. It consists of a set B with a reflexive and transitive order relation \leq , denoted by (B, \leq) . This preorder corresponds to a category with elements $b \in B$ as objects and with arrows $b_1 \rightarrow b_2$, for all elements $b_1, b_2 \in B$, where $b_1 \leq b_2$ holds in the preorder (B) . Based on this structure we can define the category \mathcal{PO} of preorders. Evidently, the objects of this category are preorders. The morphisms between these objects are monotone functions which are order-preserving.

Now, let us define a so-called forgetful functor $F : \mathcal{PO} \rightarrow \mathcal{S}$, defined by $(B, \leq) \mapsto B$. Informally, the forgetful functor takes an arbitrary object from the category \mathcal{PO} and *forgets* the additional structure defined in this particular preorder, and maps it to the carrier set of this order. We can show that this functor has a left and a right adjoint.

The left adjoint functor $L : \mathcal{S} \rightarrow \mathcal{PO}$ takes an arbitrary set B and equips it with the equality relation, i.e. $A \mapsto (A, =)$ where the equality relation is defined by $\{(a, a) | a \in A\}$. Now, due to equality we can define any function $f : A \rightarrow X$ and obtain a monotone function $(A, Eq(A)) \rightarrow (X, \leq)$. Together with f we obtain a trivial bijective correspondence between $(A, Eq(A)) \xrightarrow{f} (X, \leq)$ in the category \mathcal{PO} and $A \xrightarrow{f} X$ in the category of sets \mathcal{S} . As $L(A) = (A, Eq(A))$ and $F((X, \leq)) = X$, we obtain an adjunction $L \dashv F$, i.e. L is indeed a left-adjoint to F .

Equivalently, the right adjoint functor to F should map the objects of the category of sets \mathcal{S} into the category of preorders \mathcal{PO} . However, this time we have to define a relation which is *almighty*, i.e. all other relations which are mapped to this relation can be mapped using a monotone function. Thus, we define an *auxiliary relation* as the set $\top_{A \times A} = \{(a, a') | a, a' \in A\}$. We use the \top symbol as this relation is equivalent to a tautology (see also [1]) because it is always satisfied. As a consequence any function $X \rightarrow A$ is automatically monotone in $(X, \leq) \rightarrow (A, \top_{A \times A})$. We again obtain a rather trivial correspondence which shows that R is right adjoint to U . This time $X \xrightarrow{f} A$ defined in \mathcal{S} with $X = U((X, \leq))$ and $(X, \leq) \xrightarrow{f} (A, \top_{A \times A})$ with $R(A) = (A, \top_{A \times A})$ are in bijective correspondence with each other.

Categorical Specification of Behaviour The categorical theories introduced above provide sufficient knowledge to discuss an important application of adjunctions: the behaviour-realisation. Therefore, this section will first introduce a mechanism which allows the modelling of automata using F -algebras and gives a definition for the behaviour of automata. These definitions induce appropriate categories of automata and behaviour. On these categories we apply the concept of an adjunction.

Coalgebraic Definition of a Deterministic Automaton While automata are used as part of a rather simple modelling methodology, they often form the basis for many complex systems or theories such as parallel and distributed computing, circuit theory, language, and complexity theory. Due to their fundamental character we present their representation as a coalgebra in this section. This is basically an extension of the presentation given in the brief introduction in Section 5.1. There, we have already mentioned that coalgebras often serve as a means to describe dynamical systems. Therefore the carrier set mentioned in Section 5.3 is often referred to as the state space. We adopt this terminology as it simplifies the presentation.

First we consider the regular deterministic automaton we are already familiar with, defined through its state space Q and an input alphabet Σ . As we are talking about an automaton which models a machine or system we rename Σ to A . Any symbol in A transfers the deterministic automaton into another state. Thus, any element can also be seen as an action. Therefore we also adopt A as the set of actions in the automaton. The state transitions are defined by $\lambda \times A \rightarrow Q$. Finally, we are missing an F of all terminal states and the initial state $x_0 \in Q$.

From Section 5.3 we already know that we can transform the algebraic notion of $\lambda : Q \times A \rightarrow Q$ into the coalgebraic representation

$$Q \rightarrow Q^A.$$

Further, we translate the set of final states into a characteristic function $Q \rightarrow \{0, 1\}$ which takes the value 1 if state $x \in F$, 0 otherwise. With our knowledge from above we can now combine these functions into a function

$$Q \rightarrow Q^A \{0, 1\}.$$

Additionally, we can see that deterministic automata are models of the F -coalgebra where the functor F is the simple polynomial functor $id^A \times \{0, 1\}$. As a finite state of an automaton is an observable state, we

can also replace the binary set $\{0, 1\}$ by a larger set B which we call the output set. As a result we obtain a special type of deterministic automaton represented as the coalgebra

$$Q \xrightarrow{\langle \lambda, \delta \rangle} Q^A \times B$$

of the functor $id^A \times O$. This is the model of Mealy machines [107], an automaton model with input and output. This automaton is defined by $\langle \lambda, \delta \rangle : Q^A \times B$ where $\lambda : Q^A \rightarrow Q^A$ defines the transition function and $\delta : Q \rightarrow B$ the output or, in coalgebraic terms, the observation function. You may wonder why this informal definition does not put any restrictions on Q . Coalgebraic definitions do not require this restriction. This is due to the implicit characteristic of coalgebraic modelling. We cannot assume or determine a finite state space as it is constructed during the *observation* of the defined system. Thus, we have no finite state space as we can not derive from the observed states which state transitions the system is going to take next and thus in the future. But what do we mean by observation? As defined above, our automaton produces in each state some kind of output which can be observed. Thus, in state x we can observe $\delta(x) \in B$. Accordingly, if this state is changed by an action $a_1 \in A$ then we can observe the next output $\delta(\lambda(x)(a_1))$, etc. Thus, any finite sequence $\langle a_1, a_2, \dots, a_n \rangle \in A^*$ of actions starting from a state x corresponds to an output $\delta(\lambda(\dots\lambda(x)(a_1)\dots)(a_n))$. This is basically everything we can observe about the *behaviour* of the system starting in the unspecified state x . Thus, we specify the behaviour of our model starting in a state x as a function $beh(x) : A^* \rightarrow B$ defined by

$$beh(x) = \delta(\lambda^*(x, \sigma)), \quad \text{with } \sigma \in A^*. \quad (76)$$

λ^* is the iterated transition function which determines the non-direct successor of x by applying the single actions of the sequence $\sigma \in A^*$ consecutively. Thus, depending on the input sequence σ we can obtain many different behaviours. This discussion has retraced to some extent Rhodes's definitions and concepts discussed in Section 3.3, but couched in the formalism of category theory. We continue in the same vein.

Behaviour-Realisation We will now see why the concepts discussed above are important to discuss another notion of behaviour that it is so important for our work. This notion was developed by Goguen. He investigated mathematical system theory using category theory and defined an adjunction which he called *behaviour-realisation*. Along the lines of Jacobs (see also [88]) we will try to explain how the behaviour-realisation functor can be applied to the coalgebraic notion of a deterministic automaton which we have just introduced in Section 5.3.

As we remember from Section 5.3, adjunctions were defined between functors which are again defined between categories. Thus, in order to define a behaviour-realisation we need two categories. As expected, these categories deal with the deterministic automaton and its behaviour. Therefore, we define the category \mathcal{DA} of deterministic automata on the one hand and the category of behaviour of deterministic automata \mathcal{DB} .

The objects in category \mathcal{DA} are deterministic automata $\langle \lambda, \delta \rangle : Q \rightarrow Q^A \times B$ with some initial state $x_0 \in Q$. Consequently, the arrows between the objects are morphisms from $\langle Q \xrightarrow{\langle \lambda, \delta \rangle} Q^A \times B, x_0 \in Q \rangle$ to $\langle Q' \xrightarrow{\langle \lambda', \delta' \rangle} Q'^A \times B', x'_0 \in Q' \rangle$. These morphisms consist of the three functions,

$$\begin{array}{ccc} A & \xleftarrow{f} & A' \\ B & \xrightarrow{g} & B' \\ Q & \xrightarrow{h} & Q' \end{array}$$

which satisfy

$$\lambda'(h(x))(a') = h(\lambda(x)(f(a'))), \quad (77)$$

$$\delta'(h(x)) = g(\delta(x)), \quad \text{and} \quad (78)$$

$$h(x_0) = x'_0, \quad (79)$$

for all $a' \in A'$ and $x \in Q$. Thus, a triple (f, g, h) represents an arrow in our category. Composition over two arrows, (f_1, g_1, h_1) and (f_2, g_2, h_2) is defined by $(f_1 \circ f_2, g_2 \circ g_1, h_2 \circ h_1)$.

From Section 5.3, we already know that Equations 77 and 78 show that h is a coalgebra homomorphism. It transforms the deterministic automaton $\langle id_Q^f \circ \lambda, g \circ \delta \rangle : Q \rightarrow Q^{A'} \times B'$ into the automaton $\langle \lambda', \delta' \rangle : Q \rightarrow Q^{A'} \times B'$.

Similarly, we define the category \mathcal{DB} . Of course, the objects of this category are functions $\varphi : A^* \rightarrow B$ which assign an action sequence to an observable behaviour. Similarly to the morphisms of machines we can also define morphisms on behaviours. As behaviour is defined by functions which map a certain state into an observable output, we can simply reduce our machine morphism and use it for a morphism of behaviours. Thus, a morphism $beh \rightarrow beh'$ which transforms $A^* \xrightarrow{\varphi} B$ into $A'^* \xrightarrow{\psi} B'$ can be reduced to the functions f and g with

$$\begin{array}{ccc} A & \xleftarrow{f} & A' \\ B & \xrightarrow{g} & B' \end{array}$$

for which the condition $g(\varphi(f^*(\sigma))) = \psi(\sigma)$ for all sequences $\sigma \in A'^*$. With these functions we can also build a category with arrows (f, g) and the identity arrows (id, id) . Composition for these arrows is defined by $(f_2, g_2) \circ (f_1, g_1) = (f_1 \circ f_2, g_2 \circ g_1)$. We call this category the category of deterministic behaviour, \mathcal{DB} .

With these constructions we obtain the categories \mathcal{DA} and \mathcal{DB} . The question is, now, whether we can connect these categories since they are obviously interdependent concepts. We can define two functors which transform one category into the other, and this already connects the categories. However, we need a stronger connection which makes these functors interdependent, in order to link behaviour to automata and vice versa.

So, we start with defining the corresponding functors. As we mentioned before, the morphisms transforming two automata and two behaviours, respectively, are very similar. Thus, the functor \mathcal{E} from the category \mathcal{DA} of deterministic automata to the behaviour category \mathcal{DB} , which Goguen calls *external behaviour* [101], maps the triple (f, g, h) to (f, g) , i.e. $\mathcal{E}(f, g, h) = (f, g)$. Conversely, we can transform a behaviour to its deterministic automaton, i.e. its *realisation*, using the functor $\mathcal{R} : \mathcal{DB} \rightarrow \mathcal{DA}$. Thus, a behaviour $\psi : A^* \rightarrow B$, needs to be mapped into an automaton. This automaton is expressed by

$$B^{A^*} \longrightarrow (B^{A^*})^A \times B.$$

Why is this an automaton? We can simply define F -coalgebras on the polynomial functor $id^A \times B$. These coalgebras correspond to deterministic automata. Thus, we can also write $\mathcal{R}(f, g) = (f, g, g^{f^*})$.

It can be shown, Goguen provides a complicated but clear proof in [101], that the functor of external behaviour \mathcal{E} is right-adjoint to the realisation \mathcal{R} , i.e. $\mathcal{E} \dashv \mathcal{R}$. Thus, we obtain a direct relation between an automaton structure and its associated behaviour.

Conclusion Section 5.2 finished with an informal presentation of how an automaton can be interpreted as an algebra and how the internal structure, i.e. states and their transitions, can be separated from the input and output of the machine, i.e. from its behaviour. The construction thus presented still preserved the machine properties by linking both structures with an appropriate functor. This section refined the informal presentation of this idea by introducing the concept of polynomial functors which can be used to describe general automata. In fact, they are a comfortable means to describe the signature of modules. Mechanisms on how to transform and combine these signatures were introduced. With the introduction of the highly complex concept of adjunctions, it was then possible to link the internal structure of an automaton with its behaviour.

The power and implications of these constructions are currently understood at the level of their very basic notions. However, these observations are extremely important for our research. They help to understand in which way the particular algebraic structures we obtain from the biological systems considered can be mapped into a set of automata classes which show a particular behaviour. Additionally, with specific refinements of the fundamental theory of adjunctions, developed by Jacobs [56], whole categories

of processes can be developed. They allow for the definition of operators which enable the modelling of parallel processes. Together with an understanding of so-called feedback operators we approach a level of abstraction at which it could be possible to model the rather complex service hierarchies in BIONETS and analyse their corresponding behaviour. In short, with the understanding of this specialised theory of adjunctions, we will be able to analyse more complex systems, such as hierarchical service compositions omnipresent in BIONETS. This knowledge should allow us in the future to analyse, as well as build, complex compositions for which we have sufficient algebraic knowledge. In our case, we envisage the ability to derive a connection from a composition to its external behaviour. This is an important step towards the development of our specification language as certain behavioural components can be identified to be essential primitives of our specification language. Whether this is also possible for security characteristics has to be investigated using some primitive examples. In the next section we discuss some examples.

6 An Example of Adjunctions in Category Theory

In the following we show an important example for adjunctions in order to help the reader grasp the concept better. This section is more advanced and brings together concepts from group theory and category theory. We continue with the ‘iterative’ style, meaning that some of the concepts presented have already been discussed in previous sections, but we revisit them here at a greater depth.

6.1 Natural transformations

Natural transformations are very important in the concept of adjunctions. We remind the reader that if \mathcal{C} and \mathcal{D} are two categories and $F, G: \mathcal{C} \rightarrow \mathcal{D}$ are two functors, then τ is a *natural transformation* of F to G (usually denoted by $\tau: F \overset{\bullet}{\rightarrow} G$) if and only if for every object $C \in \mathcal{C}$ there is an arrow $\tau_C: F(C) \rightarrow G(C)$ such that if $f: A \rightarrow B$ in the category \mathcal{C} then $\tau_B \circ F(f) = G(f) \circ \tau_A$.

Basically a natural transformation transforms a functor into another in a way that preserves the morphisms. If τ is a natural transformation, which is an isomorphism, then we call it *natural isomorphism*. ‘Naturality’ in this context means that the transformation is defined in a universal way for every object in the category \mathcal{C} , i.e. only uses the arrows, and does not use any ‘internal’ properties of the structure of the objects. To understand this concept better, we show an example for a non-natural and for a natural isomorphism. This is a mathematically more rigorous presentation of concepts discussed also in Chapter 2 of [4].

Let \mathcal{V} denote the category of finite-dimensional vector spaces of the real numbers \mathbb{R} , i.e. its objects are the finite dimensional \mathbb{R} -vector spaces and the morphisms are the linear transformations from one vector space to the other. For a vector space V let V^* be the dual vector space, i.e. the vector space which contains all the linear maps from V to \mathbb{R} :

$$V^* = \{f: V \rightarrow \mathbb{R} : f \text{ is linear}\}. \quad (80)$$

It is easy to see that V^* is indeed an \mathbb{R} -vector space: the sum of two maps f_1, f_2 is defined as $(f_1 + f_2)(v) = f_1(v) + f_2(v)$ ($v \in V$) and for arbitrary $\lambda \in \mathbb{R}$ we can define $\lambda \cdot f$ as $(\lambda \cdot f)(v) = \lambda \cdot f(v)$. Moreover, if $\varphi: U \rightarrow V$ is a linear map, then $\varphi^*: V^* \rightarrow U^*$, $\varphi^*(f)(u) = f(\varphi(u))$ is a linear map. Thus $*$: $\mathcal{V} \rightarrow \mathcal{V}$ is a (contravariant) functor. Applying it twice, we obtain a covariant functor $**$: $\mathcal{V} \rightarrow \mathcal{V}$, called the double dual functor.

Now, because $*$ is a contravariant functor, it cannot be naturally isomorphic to the identity functor of \mathcal{V} . Nevertheless, $V \simeq V^{**}$ for every $V \in \mathcal{V}$. Let $B = \{b_1, \dots, b_n\}$ be a basis for V , then for every $v \in V$ there exist unique $\lambda_1, \dots, \lambda_n$ such that $v = \sum_{i=1}^n \lambda_i b_i$. We represent it with the column vector

$$[v]_B = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}. \quad (81)$$

Using this basis, we can define a basis B^* in V^* , because if $f: V \rightarrow \mathbb{R}$ is linear, then using the linearity we have $f(v) = \sum_{i=1}^n \lambda_i f(b_i)$. Thus f is determined uniquely by its values on b_1, \dots, b_n . Let $f_1, \dots, f_n \in V^*$ be linear functions such that for every $1 \leq i \neq j \leq n$ we have $f_i(b_j) = 0$ and $f_i(b_i) = 1$. Then $B^* = \{f_1, \dots, f_n\}$ is basis in V^* : if $f \in V^*$ has the property $f(b_i) = \mu_i$, then $f = \sum_{i=1}^n \mu_i f_i$ and we can denote it by the row vector $[f]_{B^*} = [\mu_1, \dots, \mu_n]$. Moreover, the value of $f(v)$ is the matrix-product of $[f]_{B^*}$ and $[v]_B$: $f(v) = [f]_{B^*} \cdot [v]_B = \sum_{i=1}^n \lambda_i \mu_i$. Now, $V \simeq V^*$ by $v \mapsto f$ if and only if their matrices in this basis are transposed to each other: $[f]_{B^*} = [v]_B^T$.

Nevertheless, this representation depends on the basis B we chose at the beginning. Let C be another basis, say $C = \{c_1, c_2, \dots, c_n\}$. Let $c_1 = 1/2b_1$, and let $c_i = b_i$ for $2 \leq i \leq n$. Now, if $v = \sum_i \lambda_i b_i$, then $v = 2\lambda_1 c_1 + \sum_{i=2}^n \lambda_i c_i$. Now, let $f_1, \dots, f_n, g_1, \dots, g_n \in V^*$ be the linear functions such that $f_i(b_i) = g_i(c_i) = 1$ and $f_i(b_j) = g_i(c_j) = 0$ ($1 \leq i \neq j \leq n$). It is easy to see that $g_1 = 2f_1$, $g_i = f_i$ for $2 \leq i \leq n$. Thus the corresponding $f_B \in V^*$ function for v is $f_B = \sum_{i=1}^n \lambda_i f_i$ using the basis B , and is $f_C = 2\lambda_1 g_1 + \sum_{i=2}^n \lambda_i g_i = 4\lambda_1 f_1 + \sum_{i=2}^n \lambda_i f_i$ using the basis C . Hence it is quite clear that in general $f_B \neq f_C$.

Now, we claim that $**$ is naturally isomorphic with the identity functor. For this we need to define $\tau_V: V \rightarrow V^{**}$ for every $V \in \mathcal{V}$. For every $v \in V$ let $\tau_V(v)$ be the linear function $g_v: V^* \rightarrow \mathbb{R}$, for which $g_v(f) = f(v)$. Thus τ_V is indeed well-defined, and it is easy to see that it is injective: if $v \neq w$, then there exists $f \in V^*$ such that $f(v) \neq f(w)$. Then g_v and g_w obtain different values on f : $g_v(f) = f(v) \neq f(w) = g_w(f)$. Because $*$ preserves the dimension, the dimensions of V and V^{**} are the same, thus τ_V is not only injective, but bijective, and therefore an isomorphism. All we need to check now, is that if $\varphi: U \rightarrow V$ is a linear map, then $\tau_U \circ \varphi = \varphi^{**} \circ \tau_V$. Let $u \in U$, then $\varphi(u) \in V$, and thus $\tau_U \circ \varphi(u)$ is the function $g_{\varphi(u)} \in V^{**}$ for which if $f \in V^*$ is arbitrary, then $g_{\varphi(u)}(f) = f(\varphi(u))$. On the other hand, $\tau_V(u)$ is the function $g_u \in U^{**}$ for which if $f' \in U^*$ is arbitrary, then $g_u(f') = f'(u)$. We thus have to understand $\varphi^{**}(g_u) \in V^{**}$. By definition, $\varphi^{**}(g_u) \in V^{**}$ has the property that if $f \in V^*$, then $\varphi^{**}(g_u)(f) = g_u(\varphi^*(f))$. Here $\varphi^*(f) \in U^*$ such that if $u' \in U$ is arbitrary, then $\varphi^*(f)(u') = f(\varphi(u'))$. Thus $g_u(\varphi^*(f)) = \varphi^*(f)(u) = f(\varphi(u))$, which is exactly what we wanted.

The main difference between the two given isomorphisms (i.e. between $V \simeq V^*$ and $V \simeq V^{**}$) is that for proving $V \simeq V^*$ we needed to use the internal theory of finite dimensional vector spaces and their linear maps. For proving $V \simeq V^{**}$ we did not need this internal theory, we only needed to use the basic definitions and the category theoretical notions.

Now we move to construct an example for adjoint functors. This example will show how to create free groups as the adjunction to the forgetful functor. For this first we need to understand the concept of a free group.

6.2 Free groups

The idea behind the free group construction is basically finding a ‘biggest’ (or ‘most general’) group generated by a set X . Being a bit more precise, a free group generated by X (let us denote it by F_X) is a group which contains X as a subset and contains everything which it ‘is forced to contain’ by containing X . E.g. if x and y are two elements of X , then F_X has to contain their product xy , their inverses x^{-1} and y^{-1} , and it has to contain some arbitrary product of x and y , like $x^2 y^3 x^{-1} y x$. By ‘biggest’ we mean that none of these ‘enforced’ elements coincide, unless they have to by the group axioms. E.g. the elements $(xy)^{-1}$ and $y^{-1} x^{-1}$ have to be equal, because for every group $(xy)^{-1} = y^{-1} x^{-1}$. Similarly, x and $xx^{-1} x$ will be equal in F_X , but x and xy will be two distinct elements in F_X .

Free constructions are quite important in algebra, one can construct free rings or free semigroups, etc. For some particular algebras the concept of ‘biggest’ can be really hard to fathom in a constructive way; therefore, we define this notion by the so-called ‘universal property’. This basically says that F_X is a free group if for any group G and any function $f: X \rightarrow G$ we can extend f on the whole F_X to a group homomorphism.

Definition 13. Let X be a set. We call F_X the *free group generated by X* iff X generates F_X as a group, and for every group G and for every function $f: X \rightarrow G$ there exists a group homomorphism $f^*: F_X \rightarrow G$ such that $f^*|_X = f$.

It is not at all clear that for every set X the free group F_X exists. What follows from the definition is that if F_X exists, then it is unique. It is a corollary of the observation that an $X \rightarrow G$ map extends uniquely over F_X :

Assertion 14. Let X be a set and let F_X be a free group generated by X . Let G be an arbitrary group. Let $\varphi, \psi: F_X \rightarrow G$ homomorphisms such that $\varphi|_X = \psi|_X$. Then $\varphi = \psi$.

Proof: Let H be the subset of F_X for which $\varphi = \psi$:

$$H = \{h \in F_X \mid \varphi(h) = \psi(h)\}.$$

Then it is easy to see that H is a subgroup, because φ and ψ are homomorphisms. Since H contains X , it contains the smallest subgroup in F_X generated by X , which is F_X itself. Thus $H = F_X$, hence $\varphi = \psi$ over F_X . ■

Assertion 15. Let X be a set. If F_X and F'_X are both free groups generated by X , then $F_X \simeq F'_X$.

Proof: By the definition there exist homomorphisms $\varphi: F_X \rightarrow F'_X$ and $\psi: F'_X \rightarrow F_X$ which extend the identity on the set X . Now, $\psi \circ \varphi$ is the identity morphism on F_X because of Assertion 14. Similarly, $\varphi \circ \psi$ is the identity on F'_X . This proves that φ and ψ are isomorphisms between F_X and F'_X ; moreover, they are inverses of each other. ■

Assertion 16. Let X be a set. Then F_X exists.

Proof: [only sketch of the proof] Let F be all the words built up by the elements from $X \cup X^{-1}$. Let us define an *elementary change* in a word: an elementary change is made when, for some $x \in X \cup X^{-1}$, we either add xx^{-1} to the word or remove xx^{-1} from the word. We say that two words s and t are equivalent ($s \sim t$) iff we can obtain s from t by elementary changes. It can be proven that \sim is an equivalence relation on F . Thus we can consider the set of equivalence classes F/\sim . It can be proven that $F_X \simeq F/\sim$. The proof is quite technical and we omit it. The reader may refer to [108] for the remainder of the proof. ■

Free groups are important not only in Group theory but in other parts of mathematics, e.g. in Topology. It is a direct consequence of the definition that if a group G is generated by a set of elements X , then G is a factor of F_X . Earlier we saw that by Cayley's theorem every group is a subgroup of a symmetric group; the abovementioned claim shows that every group is a quotient of a free group. We do not continue on the theory of free groups; we only wanted to demonstrate their importance to motivate our example for adjunctions in category theory.

6.3 Left-adjoint to the 'forgetful' functor

Let \mathcal{SET} be the category of sets and let \mathcal{GROUP} be the category of groups. Let X be an arbitrary set and let F_X be the free group generated by the set X .

Let $G: \mathcal{GROUP} \rightarrow \mathcal{SET}$ be the 'forgetful' functor, which forgets the group structure, i.e. for every group H the functor G sends H to its base set $G(H)$ and every group homomorphism to the corresponding function over the base sets. This functor is arguably one of the most basic functors in category theory.

We show that G admits a left-adjoint functor $F: \mathcal{SET} \rightarrow \mathcal{GROUP}$ which is the 'free' functor: for every set X the group $F(X)$ will be the free group F_X generated by X and, if $f: X \rightarrow Y$ is a function, then $F(f): F_X \rightarrow F_Y$ is the homomorphism which extends $f: X \rightarrow Y$. Then F is left-adjoint to G .

Natural isomorphism of $\text{hom}_{\mathcal{GROUP}}(F(X), H) \simeq \text{hom}_{\mathcal{SET}}(X, G(H))$ Maybe the easiest to see is that $\text{hom}_{\mathcal{GROUP}}(F(X), H) \simeq \text{hom}_{\mathcal{SET}}(X, G(H))$ via a natural isomorphism Φ . Let X be an arbitrary set and H be an arbitrary group. Then $F(X)$ is the free group generated by X . Now, every $\varphi: F(X) \rightarrow H$ homomorphism is uniquely determined (Assertion 14) by $\varphi|_X$. Of course, $\varphi|_X$ can be considered as a function from X to the base set of H , i.e. a morphism $X \rightarrow G(H)$. Let $\Phi_{X,H}: \text{hom}_{\mathcal{GROUP}}(F(X), H) \rightarrow \text{hom}_{\mathcal{SET}}(X, G(H))$ be this transformation. It is clear then that $\Phi_{X,H}(\varphi)$ exists. It is invertible (hence it is an isomorphism), because if $\psi: X \rightarrow G(H)$, then ψ is a function from X to the base set of H . By the definition of the free group and by Assertion 14, it can be uniquely extended to a group homomorphism $\psi^*: F_X \rightarrow H$, for which $\psi^*|_X = \psi$. This means exactly that $\Phi_{X,H}$ is invertible.

What remains to be seen is that $\Phi_{X,H}$ is a *natural* isomorphism (from which the naturality of $\Phi_{X,H}^{-1}$ follows easily). By definition we need to prove that the diagram in Fig. 36 commutes.

That is for every function $f: X' \rightarrow X$ and for every group homomorphism $g: H \rightarrow H'$ and for every $\varphi: F(X) \rightarrow H$ we have

$$\Phi_{X',H'}(g \circ \varphi \circ F(f)) = G(g) \circ \Phi_{X,H}(\varphi) \circ f. \quad (82)$$

$$\begin{array}{ccc}
 \text{hom}_{\mathcal{G}\mathcal{R}\mathcal{O}\mathcal{U}\mathcal{P}}(F(X), H) & \xrightarrow{\Phi_{X,H}} & \text{hom}_{\mathcal{S}\mathcal{E}\mathcal{T}}(X, G(H)) \\
 \downarrow g \circ _ \circ F(f) & & \downarrow G(g) \circ _ \circ f \\
 \text{hom}_{\mathcal{G}\mathcal{R}\mathcal{O}\mathcal{U}\mathcal{P}}(F(X'), H') & \xrightarrow{\Phi_{X',H'}} & \text{hom}_{\mathcal{S}\mathcal{E}\mathcal{T}}(X', G(H'))
 \end{array}$$

Fig. 36. Naturality square for $\Phi_{X,H}$: $\text{hom}_{\mathcal{G}\mathcal{R}\mathcal{O}\mathcal{U}\mathcal{P}}(F(X), H) \simeq \text{hom}_{\mathcal{S}\mathcal{E}\mathcal{T}}(X, G(H))$

For proving it let $x' \in X'$ be arbitrary and let us see the left hand side. Now, $f(x') \in X$, the functor F only embeds it into the free group $F(X)$. The homomorphism φ sends $F(f)(x')$ to H , then g moves it to H' . Essentially $\Phi_{X',H'}$ then forgets about the group structure, and embeds all the morphisms into the category $\mathcal{S}\mathcal{E}\mathcal{T}$.

Since $f(x') \in X$, we have $\varphi(F(f)(x')) = \varphi|_X(f(x')) = \Phi_{X,H}(\varphi)(f(x'))$. Now, looking at the right hand side, composing $\Phi_{X,H}(\varphi)(f(x'))$ by g moves it to H' , and the functor G only forgets about the group structure.

Thus for every $x' \in X'$ we have

$$\Phi_{X',H'}(g(\varphi(F(f))))(x') = G(g)(\Phi_{X,H}(\varphi)(f(x'))), \quad (83)$$

which proves (82), and finishes the proof that F is left adjoint to G .

Unit, counit The counit $\varepsilon: FG \xrightarrow{\bullet} 1_{\mathcal{G}\mathcal{R}\mathcal{O}\mathcal{U}\mathcal{P}}$ and the unit $\eta: 1_{\mathcal{S}\mathcal{E}\mathcal{T}} \xrightarrow{\bullet} GF$ are natural transformations such that the composition $F \xrightarrow{F\eta} FGF \xrightarrow{\varepsilon F} F$ is the identity transformation 1_F and the composition $G \xrightarrow{\eta G} GFG \xrightarrow{G\varepsilon} G$ is the identity transformation 1_G . This is equivalent to the equations $1_F = \varepsilon F \circ F\eta$ and $1_G = G\varepsilon \circ \eta G$, which means that for every set X and for every group H we have $1_{F(X)} = \varepsilon_{F(X)} \circ F(\eta_X)$ and $1_{G(H)} = G(\varepsilon_H) \circ \eta_{G(H)}$.

In our example let H be an arbitrary group, and consider first $FG(H)$. This is the free group generated by $G(H)$, i.e. the free group generated by the elements of H . Now, let $\varepsilon_H: FG(H) \rightarrow H$ be the unique group homomorphism, which sends the generators of $FG(H)$ (i.e. the elements of H) to themselves in H . Similarly, let X be an arbitrary set and consider $GF(X)$. This is the base set of the free group generated by X . Let $\eta_X: X \rightarrow GF(X)$ be the inclusion map of the generator set X . This defines both ε and η . We prove that they are counit and unit, respectively.

First we acknowledge the triangular equations. Let H be an arbitrary group and consider

$$G(H) \xrightarrow{\eta_{G(H)}} GFG(H) \xrightarrow{G(\varepsilon_H)} G(H). \quad (84)$$

Here, $GFG(H)$ is the base set of the free group generated by the elements of H . The morphism $\eta_{G(H)}$ is the inclusion of the generator set $G(H)$ into $GFG(H)$. Now, ε_H is the homomorphism which sends the generators of $FG(H)$ to themselves in H , and G just forgets about the group structure. Thus (84) is indeed the identity on $G(H)$.

Similarly, let X be an arbitrary set and consider

$$F(X) \xrightarrow{F(\eta_X)} FGF(X) \xrightarrow{\varepsilon_{F(X)}} F(X). \quad (85)$$

Here, $FGF(X)$ is the free group generated by the base set of $F(X)$. Here, $FGF(X)$ looks the same as $F(X)$, because both of them contain the same words (which are generated by X). Nevertheless, there is an important difference, which comes from the fact that $FGF(X)$ is *freely* generated by elements of $F(X)$, and thus if $x, x' \in X$, then xx' as a generator of $FGF(X)$ is not the same as the product of x and x' (which are generators of $FGF(X)$ as well). To distinguish them we imagine that in every word of $FGF(X)$ the generators (which are elements of $F(X)$) are in parentheses, i.e. (xx') is a generator of $FGF(X)$, while $(x)(x')$ is the product of two generators of $FGF(X)$. Now, $F(\eta_X)$ is the group homomorphism, which sends $F(X)$ to $FGF(X)$ by sending every generator x of $F(X)$ to the length 1 word (x) in $FGF(X)$. Thus

$F(\eta_X)$ is basically the inclusion of $F(X)$ into $FGF(X)$ by sending every word of $F(X)$ into the same word with the same length in $FGF(X)$. Then $\varepsilon_{F(X)}$ is the homomorphism which sends the generators of $FGF(X)$ to themselves in $F(X)$, i.e. which forgets the parentheses in $FGF(X)$ (and e.g. sends both $(x)(x')$ and (xx') to xx'). Thus (85) is indeed the identity on $F(X)$.

It remains to be proved that both ε and η are natural. Let X and X' be arbitrary sets and let $f: X' \rightarrow X$ be arbitrary. Then naturality of η means that the diagram in Fig. 37 commutes.

$$\begin{array}{ccc} X' & \xrightarrow{\eta_{X'}} & GF(X') \\ \downarrow f & & \downarrow GF(f) \\ X & \xrightarrow{\eta_X} & GF(X) \end{array}$$

Fig. 37. Naturality square for η

Let $x' \in X'$ be arbitrary. Then $f(x') \in X$, which is embedded into $GF(X)$ by η_X as a generator. On the other hand, $\eta_{X'}$ embeds x' into $GF(X')$ as a generator. Then $GF(f)$ sends every generator of $F(X')$ into a generator of $F(X)$ by f , i.e. sends $\eta_{X'}(x')$ into $f(x')$ embedded into $GF(X)$ as a generator. Thus η is clearly a natural transformation.

Now, let H and H' be arbitrary groups and let $g: H \rightarrow H'$ be an arbitrary homomorphism. Then naturality of ε means that the diagram in Fig. 38 commutes. Let $h \in H$ be arbitrary. Then ε_H sends h as

$$\begin{array}{ccc} FG(H) & \xrightarrow{\varepsilon_H} & H \\ \downarrow FG(g) & & \downarrow g \\ FG(H') & \xrightarrow{\varepsilon_{H'}} & H' \end{array}$$

Fig. 38. Naturality square for ε

a generator of $FG(H)$ into h , which is sent into $g(h)$. On the other hand, $FG(g)$ sends every generator of $FG(H)$ into a generator of $FG(H')$ via g , i.e. h is sent to $g(h)$. Then $\varepsilon_{H'}$ sends a generator of $FG(H')$ into itself in H' , i.e. $g(h)$ is sent to $g(h)$. Thus ε is clearly a natural transformation.

6.4 Further notes and examples

Maybe understanding adjunctions is the easiest via the natural isomorphism of hom-sets. The functor $F: \mathcal{D} \rightarrow \mathcal{C}$ is left adjoint to $G: \mathcal{C} \rightarrow \mathcal{D}$ if and only if for every $C \in \mathcal{C}$, $D \in \mathcal{D}$ we have $\text{hom}_{\mathcal{C}}(F(D), C) \simeq \text{hom}_{\mathcal{D}}(D, G(C))$ in a natural way. This means the following: the functor F projects the category \mathcal{D} into the category \mathcal{C} and the functor G projects the category \mathcal{C} into the category \mathcal{D} such that the arrows from $F(\mathcal{D})$ to \mathcal{C} correspond to the arrows from \mathcal{D} to $G(\mathcal{C})$. It is somewhat like finding an inverse to the functor, except that FG and GF need not be the identity of the corresponding categories, only naturally isomorphic to the identity.

Adjunctions can be seen as the ‘most universal’ solutions to a certain problem. Our free group example can be seen as an example for this argument: $G: \mathcal{GRUUP} \rightarrow \mathcal{SET}$ is the most basic functor, which makes every group into a set. Its adjoint F is the most universal way to make a group from every set, that is if X is a set, then $F(X)$ will be a group which has the properties X had in the category \mathcal{SET} , extended by further properties so that it becomes a group. The free group $F(X)$ is universal in the sense that no other identities hold than those which are imposed by being a group. Phrasing the universality of

F in another way is: for every set $X \in \mathcal{SET}$ and for every group $H \in \mathcal{GROUP}$ every homomorphism $f: X \rightarrow G(H)$ in \mathcal{SET} uniquely extends to a homomorphism $g: F(X) \rightarrow H$ in \mathcal{GROUP} .

Adjunctions can give rise to many constructions, especially in algebra, if we try to find left adjoints to different forgetful or inclusion functors. Some more examples might help the reader to further appreciate the importance of this category theoretical notion.

1. Let \mathcal{RING} denote the category of unital rings (which have a multiplicative identity), and let \mathcal{RNCG} denote the not necessarily unital rings. Let $G: \mathcal{RING} \rightarrow \mathcal{RNCG}$ be the forgetful functor (forgetting about the identity). Then G has a left adjoint functor $F: \mathcal{RNCG} \rightarrow \mathcal{RING}$, which is the universal construction of adding an identity to a ring that does not necessarily have one. For a ring $R \in \mathcal{RNCG}$ which might not have an identity $F(R) \in \mathcal{RING}$ will be the ring with elements $n + r$. Thus the functor F adjoins an identity to every ring and adjoins every element that has to be in the new ring due to the fact that it has an identity. This is optimal in the sense that it adjoins an identity and some more elements, but exactly those which are imposed by the identity and the original structure. It has the universal property: if $R \in \mathcal{RNCG}$, $S \in \mathcal{RING}$, then every $f: R \rightarrow G(S)$ homomorphism in \mathcal{RNCG} uniquely extends to a homomorphism $g: F(R) \rightarrow S$ in \mathcal{RING} .
2. Let \mathcal{CRING} denote the category of commutative unital rings and let \mathcal{CRING}^* denote the category of commutative unital pointed rings, i.e. the objects are pairs (R, r) for some $r \in R$, and the morphisms are ring homomorphisms preserving the distinguished element. Then the forgetful functor $G: \mathcal{CRING}^* \rightarrow \mathcal{CRING}$ has a left adjoint $F: \mathcal{CRING} \rightarrow \mathcal{CRING}^*$, which maps a unital commutative ring R to the pair $(R[x], x)$, where $R[x]$ is the polynomial ring. Again this is an example of a free construction: we adjoin a new element x to R in a free or universal way.
3. Let \mathcal{FIELD} denote the category of fields, and let \mathcal{DOM} denote the category of integral domains (i.e. commutative unital rings without zero divisors). The forgetful functor $G: \mathcal{FIELD} \rightarrow \mathcal{DOM}$ has a left adjoint $F: \mathcal{DOM} \rightarrow \mathcal{FIELD}$, which is the well-known quotient field construction. Again, this extends an integral domain with new elements in a universal way, so that it becomes a field.
4. Let \mathcal{AB} denote the category of Abelian groups, and consider the inclusion functor $G: \mathcal{AB} \rightarrow \mathcal{GROUP}$. Then it admits a left adjoint $F: \mathcal{GROUP} \rightarrow \mathcal{AB}$, for which $F(H) = H/H'$ ($H \in \mathcal{GROUP}$), the quotient by the commutator subgroup, which is the biggest Abelian factor of the group H .

Finally we mention that behaviour is left-adjoint to realisation. This sentence of course needs precise formulations, which can be found in [101] or that we have already discussed briefly in Sections 3.3 and 5.3. The main idea is that finite-state automata form a category, and their behaviour (which is the input-output function they compute) forms a category, as well. Now, a functor G from the category automata to the category of behaviour which assigns to every automaton its behaviour admits a left-adjoint functor F . This functor assigns to every behaviour the (in some sense) minimal automaton, which has the same behaviour. This is a well-known construction; for further reference we point the reader to [16].

7 Conclusion

In this long chapter we have tried to give a comprehensive synthesis of the main theories, concepts, and insights that we have studied and achieved in the second half of the BIONETS project. Although we have not proven any new theorems, we have made considerable progress in understanding the conceptual foundations of bio-computing and in identifying the mathematical theories that are most likely to lead to a unified formalism to represent and model cell biology, automata structure/behaviour, and specification languages. The highlights of our results are as follows:

- The fundamental building blocks of biology are Structure, Function, and Organisation. To these, we should add the underlying building block of Memory (seen ultimately as a statistical physics phenomenon, or as the “crystallisation of order”, to use a Kauffman phrase), which enabled evolution to bootstrap them from the primordial soup in the first place.
- These four building blocks carry different relative importance at different physical scales: memory processes (evolution, learning, etc) are most important at large scales characteristic of multicellular organisms. Causal and dynamic interdependence between structure and function are most important

at sub-cellular scales, whereas all four are equally important at the meso-scale of individual cells where autopoietic effects predominate.

- Biological Organisation, in the sense of operational closure and autopoiesis, maps to Software Architecture, but we have not yet reached usable conclusions in this regard that could be applied e.g. to networking or to software engineering.
- In collaboration with the Biocomputation Laboratory of the University of Hertfordshire we have proposed a methodology of analysis that, starting from the same cellular pathway, looks at the algebraic structure of its discrete automaton and its continuous ODE representation, in parallel.
- From an algebraic point of view, ‘structure’ refers to permutation groups of subsets of the state set of such an automaton, which we are calling ‘static symmetries’.
- From a dynamical systems point of view ‘function’ or ‘behaviour’ may be related to the Lie symmetries of the ODE system derived from the same cellular pathway, which we are calling ‘dynamic symmetries’.
- We have not been able to prove or disprove the hypothesis that these two kinds of symmetries may be related.
- From a coalgebraic point of view, ‘behaviour’, which can be thought of as a black-box input-output mathematical function, is expressible through coalgebra.
- The relationship between biological structure and function appears to map to the relationship between automata structure and behaviour. The latter relationship is formalised by category theory through the concept of adjunction, which provides the right setting for developing an algorithm by which one can be derived from the other. We have not developed such an algorithm yet.
- Category theory can, additionally, provide a two-way translation from automata coalgebra to the skeleton of a specification language.
- Once such a specification language has been developed, adding an operational semantics could endow a suitable run-time environment with the ability to instantiate dynamically a software artifact from the specification of its behaviour. In such a scenario, the environment is the analogue of a cell, and the dynamic instantiation process is the algorithm that mediates the causal link between structure and behaviour.

In conclusion, we believe we have laid the foundations for a biologically-inspired theory of interaction computing that promises to be general enough to enable the development of applications in computer science such as symbiotic security, new specification languages for programming at much higher levels of abstraction than is currently possible, and new modelling frameworks based on discrete mathematics and finite automata for computational systems biology.

Our next steps will be to continue to integrate our ideas with the work of great scientists such as Stuart Kauffman [41], John Rhodes [16], Robin Milner [109], and Joseph Goguen [101].

Acknowledgements

The authors are grateful to Prof C. L. Nehaniv of the University of Hertfordshire, UK, for reviewing the algebra and category theory parts of this paper. P. Dini’s time was partially supported also by the OPAALS Project EU-IST-FP6-034824.

References

1. P. Dini and D. Schreckling, “On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security,” in *Paradigms for Biologically-Inspired Autonomic Networks and Services*, E. Altman, P. Dini, D. Miorandi, and D. Schreckling, Eds., 2010.
2. L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, 1st ed. London: Springer, Nov. 2002.
3. E. Altman, P. Dini, D. Miorandi, and D. Schreckling, *D2.1.1: Paradigms and Foundations of BIONETS Research*, <http://www.bionets.eu>, 2007.
4. P. Dini, D. Schreckling, and L. Yamamoto, *D2.2.4: Evolution and Gene Expression in BIONETS: A Mathematical and Experimental Framework*. <http://www.bionets.eu>: BIONETS Deliverable, European Commission, 2008.

5. P. Dini, G. Horvath, D. Schreckling, and H. Pfeffer, *D2.2.9: Mathematical Framework for Interaction Computing with Applications to Security and Service Choreography*. <http://www.bionets.eu>: BIONETS Deliverable, European Commission, 2009.
6. J. Lahti, J. Huusko, D. Miorandi, L. Bassbouss, H. Pfeffer, P. Dini, G. Horvath, S. Elaluf-Calderwood, D. Schreckling, and L. Yamamoto, *D3.2.7: Autonomic Services within the BIONETS SerWorks Architecture*. <http://www.bionets.eu>: BIONETS Deliverable, European Commission, 2009.
7. D. Schreckling, M. Brunato, P. Dini, L. Dóra, A. Faschingbauer, J. Golic, G. Horvath, F. Martinelli, and M. Petrocchi, *D4.6: Security in BIONETS*, D. Schreckling, Ed. <http://www.bionets.eu>: BIONETS Deliverable, European Commission, 2009.
8. R. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. <http://www.ics.uci.edu/fielding/pubs/dissertation/top.htm>: UC Irvine PhD Dissertation, 2000.
9. P. Dini, G. Briscoe, I. Van Leeuwen, A. J. Munro, and S. Lain, *D1.3: Biological Design Patterns of Autopoietic Behaviour in Digital Ecosystems*. http://files.opaals.org/OPAALS/Year_3_Deliverables/WP01/: OPAALS Deliverable, European Commission, 2009.
10. P. Dini, *D18.4-Report on self-organisation from a dynamical systems and computer science viewpoint*. <http://files.opaals.org/DBE>: DBE Project, 2007.
11. S. Maeda, "The similarity method for difference equations," *IMA Journal of Applied Mathematics*, vol. 38, pp. 129–134, 1987.
12. W. D. Maurer and J. L. Rhodes, "A property of finite simple non-abelian groups," *Proc. Amer. Math. Soc.*, vol. 16, pp. 552–554, 1965.
13. G. Horvath, *Functions and Polynomials over Finite Groups from the Computational Perspective*. PhD Dissertation: The University of Hertfordshire, 2008.
14. H. K. Kaiser, "Contributions to the theory of polynomially complete algebras," *Anais da Academia Brasileira de Ciencias*, vol. 48, no. 1, pp. 1–5, 1976.
15. P. Dini and E. Berdou, *D18.1-Report on DBE-Specific Use Cases*. <http://files.opaals.org/DBE>: DBE Project, 2004.
16. J. L. Rhodes, *Applications of Automata Theory and Algebra via the Mathematical Theory of Complexity to Biology, Physics, Psychology, Philosophy, and Games*. World Scientific Press, 2009, foreword by Morris W. Hirsch, edited by Chrystopher L. Nehaniv (Original version: University of California at Berkeley, Mathematics Library, 1971).
17. G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems*, G. Nicolis and I. Prigogine, Eds. New York: Wiley, 1977.
18. D. Bjørner and C. B. Jones, Eds., *The Vienna Development Method: The Meta-Language*, ser. Lecture Notes in Computer Science, vol. 61. Springer, 1978.
19. J. M. Spivey, *The Z notation: a reference manual*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.
20. R. Milner, *A Calculus of Communicating Systems*, ser. Lecture Notes in Computer Science. Springer, 1980, vol. 92.
21. R. Milner, J. Parrow, and D. Walker, "A Calculus of Mobile Processes, I," *Inf. Comput.*, vol. 100, no. 1, pp. 1–40, 1992.
22. C. A. R. Hoare, "Communicating sequential processes," *Commun. ACM*, vol. 21, no. 8, pp. 666–677, 1978.
23. T. Bolognesi and E. Brinksma, "Introduction to the ISO specification language LOTOS," *Comput. Netw. ISDN Syst.*, vol. 14, no. 1, pp. 25–59, 1987.
24. J. A. Bergstra and J. W. Klop, "ACPt: a universal axiom system for process specification," pp. 447–463, 1989.
25. H. Andréka, I. Németi, and I. Sain, *Universal Algebraic Logic*, 1st ed., ser. Studies in Universal Logic. Springer, to appear.
26. D. Schreckling and P. Dini, "Distributed Online Evolution: An Algebraic Problem?" in *IEEE 10th Congress on Evolutionary Computation, Trondheim, Norway, 18-21 May, 2009*.
27. R. Rosen, "A Relational Theory Of Biological Systems," *Bulletin of Mathematical Biophysics*, vol. 20, pp. 245–260, 1958.
28. —, "The Representation Of Biological Systems From The Standpoint Of The Theory Of Categories," *Bulletin of Mathematical Biophysics*, vol. 20, pp. 317–341, 1958.
29. N. Rashevsky, *Mathematical Biophysics and Physico-Mathematical Foundations of Biology Vol II*, N. Rashevsky, Ed. New York: Dover, 1960.
30. A. Cornish-Bowden and M. L. Cardenas, "Self-organization at the origin of life," *Journal of Theoretical Biology*, vol. 252, pp. 411–418, 2008.
31. H. Maturana and F. Varela, *Autopoiesis and Cognition. the Realization of the Living*. Boston: D. Reidel Publishing Company, 1980.
32. R. Rosen, *Life Itself*. New York: Columbia University Press, 1991.
33. D. Chu and W. K. Ho, "A Category Theoretical Argument Against the Possibility of Artificial Life: Robert Rosens Central Proof Revisited," *Artificial Life*, vol. 12, pp. 117–134, 2006.
34. —, "The Localization Hypothesis and Machines," *Artificial Life*, vol. 13, p. 299302, 2007.
35. —, "Computational Realizations of Living Systems," *Artificial Life*, vol. 13, p. 369381, 2007.
36. A. H. Louie, "A Living System Must Have Noncomputable Models," *Artificial Life*, vol. 13, p. 293297, 2007.
37. O. Wolkenhauer, "Interpreting Rosen," *Artificial Life*, vol. 13, p. 291292, 2007.
38. R. Rosen, "Some relational cell models: The metabolism-repair systems," in *Foundations of Mathematical Biology, Vol II: Cellular Systems*, R. Rosen, Ed. Academic Press, 1972.
39. R. P. Gabriel and R. Goldman, "Conscientious software," in *OOPSLA06, Portland, Oregon, 22-26 October, 2006*.
40. M. Eigen and P. Schuster, "The Hypercycle," *Naturwissenschaften*, vol. 65, no. 1, 1978.
41. S. Kauffman, *The Origins of Order: Self-Organisation and Selection in Evolution*. Oxford: Oxford University Press, 1993.

42. A. Egri-Nagy, C. L. Nehaniv, and M. J. Schilstra, "Symmetry groups in biological networks," in *Information Processing in Cells and Tissues, IPCAT09 Conference*, Journal preprint, 5-9 April, 2009.
43. A. Egri-Nagy, P. Dini, C. L. Nehaniv, and M. J. Schilstra, "Transformation Semigroups as Constructive Dynamical Spaces," in *Proceedings of the 3rd OPAALS International Conference*, Aracaju, Sergipe, Brazil, 22-23 March, 2010.
44. K. Krohn and J. Rhodes, "Algebraic Theory of Machines. I. Prime Decomposition Theorem for Finite Semigroups and Machines," *Transactions of the American Mathematical Society*, vol. 116, pp. 450–464, 1965.
45. A. Egri-Nagy and C. L. Nehaniv, "SgpDec – software package for hierarchical coordinatization of groups and semigroups, implemented in the GAP computer algebra system," (<http://sgpdec.sf.net>), 2008.
46. K. Krohn, R. Langer, and J. Rhodes, "Algebraic Principles for the Analysis of a Biochemical System," *Journal of Computer and System Sciences*, vol. 1, pp. 119–136, 1967.
47. C. L. Nehaniv and J. L. Rhodes, "The Evolution and Understanding of Hierarchical Complexity in Biology from an Algebraic Perspective," *Artificial Life*, vol. 6, pp. 45–67, 2000.
48. A. Egri-Nagy, C. L. Nehaniv, J. L. Rhodes, and M. J. Schilstra, "Automatic Analysis of Computation in Biochemical Reactions," *BioSystems*, vol. 94, no. 1-2, pp. 126–134, 2008.
49. A. Egri-Nagy and C. L. Nehaniv, "Algebraic Properties of Automata Associated to Petri Nets and Applications to Computation in Biological Systems," *BioSystems*, vol. 94, no. 1-2, pp. 135–144, 2008.
50. —, "Hierarchical coordinate systems for understanding complexity and its evolution with applications to genetic regulatory networks," *Artificial Life*, vol. 14, no. 3, pp. 299–312, 2008, (Special Issue on the Evolution of Complexity).
51. M. Barr and C. Wells, *Category Theory for Computing Science*, C. A. R. Hoare, Ed. Hertfordshire, UK: Prentice Hall International Ltd, 1990.
52. S. Awodey, *Category Theory*. Clarendon Press, 2006.
53. J. A. Goguen, "Minimal realization of machines in closed categories," in *Bulletin of the AMS*, 1972, vol. 78, pp. 777–783.
54. —, "Discrete-time machines in closed monoidal categories. i," *Journal of Computer and System Sciences*, vol. 10, no. 1, pp. 1–43, 1975.
55. D. E. Rydeheard, "Adjunction," in *CTCS*, ser. Lecture Notes in Computer Science, D. H. Pitt, S. Abramsky, A. Poigné, and D. E. Rydeheard, Eds., vol. 240. Springer, 1985, pp. 51–57.
56. B. P. Jacobs, "Automata and behaviours in categories of processes," Amsterdam, The Netherlands, The Netherlands, Tech. Rep., 1996.
57. J. Goguen and G. Malcolm, "A hidden agenda," *Theor. Comput. Sci.*, vol. 245, no. 1, pp. 55–101, 2000.
58. B. C. Pierce, *Basic Category Theory for Computer Scientists*. Cambridge, Massachusetts: The MIT Press, 1991.
59. H. Ehrig, M. Große-Rhode, and U. Wolter, "Applications of category theory to the area of algebraic specification in computer science," in *LNCS*, Proc. WADT11, vol. 6, pp. 1–35, 1998.
60. W. R. Ashby, *An Introduction to Cybernetics*. London: Chapman & Hall Ltd., 1956. [Online]. Available: <http://pespmc1.vub.ac.be/books/IntroCyb.pdf>
61. G. M. Weinberg, *An Introduction to General Systems Thinking*. New York: Dorset House, 2001, silver Anniversary Edition.
62. H. Maturana and F. Varela, *The Tree of Knowledge. The Biological Roots of Human Understanding*. Boston and London: Shambhala, 1998.
63. L. von Bertalanffy, *General System Theory: Foundations, Developments, Applications*. New York: George Braziller, 1969.
64. H. Pfeffer, "An Underlay System for Enhancing Dynamicity within Web Mashups," *Journal on Advances in Software*, vol. 2, no. 1, 2009.
65. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell (4th Ed.)*. New York: Garland Science, 2002.
66. P. Dini and D. Schreckling, "A Research Framework for Interaction Computing," in *Proceedings of the 3rd OPAALS International Conference*, Aracaju, Sergipe, Brazil, 22-23 March, 2010.
67. I. Van Leeuwen, A. J. Munro, I. Sanders, O. Staples, and S. Lain, "Numerical and Experimental Analysis of the p53-mdm2 Regulatory Pathway," in *Proceedings of the 3rd OPAALS International Conference*, Aracaju, Sergipe, Brazil, 22-23 March, 2010.
68. D. Golding and P. Wegner, "The church-turing thesis: Breaking the myth," in *Computability in Europe (CiE) conference series*, 2005.
69. A. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. 42:2, pp. 230–265, 1936, a correction, *ibid*, 43, 1937, pp. 544-546.
70. L. Yamamoto (Ed.), "Specification of Service Life-Cycle," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.2.1), Feb. 2007.
71. —, "Specification of Service Evolution," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.2.2), July 2007.
72. J. Lahti (Ed.), "Advanced Service LifeCycle and Integration," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.2.4), July 2008.
73. P. Heiko (Ed.), "Deliverable D3.2.5: Evaluating the Fitness of Service Compositions," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D3.2.5), Technische Universität Berlin, 2008.
74. G. Winskel and M. Nielsen, *Categories in Concurrency in Semantics and Logics of Computation*. Cambridge: Cambridge University Press, 1997.
75. S. Abramsky, S. Gay, and R. Nagarajan, "Interaction Categories and the Foundations of Typed Concurrent Programming," in *Proceedings of the NATO Advanced Study Institute on Deductive program design*. New York: Springer-Verlag, 1996, pp. 35–113.

76. J. R. B. Cockett and D. A. Spooner, "SProc Categorically," in *Proceedings of 5th International Conference on Concurrency Theory*. Springer LNCS, 1994, pp. 46–159.
77. K. Worytkiewicz, "Concrete Process Categories," *Electronic Notes in Theoretical Computer Science*, vol. 68:1, pp. 70–85, 2002.
78. A. Joyal, M. Nielsen, and G. Winskel, "Bisimulation from Open Maps," *Journal of Information and Computation*, vol. 127, pp. 164–185, 1994.
79. P. Wadler, "Comprehending Monads," *Mathematical Structures in Computer Science*, vol. 2, pp. 461–493, 1992, special issue of selected papers from 6th Conference on Lisp and Functional Programming.
80. J. D. Dixon and B. Mortimer, *Permutation Groups*, 1st ed. New York: Springer GTM, 1996.
81. P. Olver, *Applications of Lie Groups to Differential Equations*. Springer, 1986.
82. N. Carter, "Group explorer – software package for the visualisation of groups," (<http://grouppexplorer.sourceforge.net>), 2008.
83. —, *Visual Group Theory*. Washington, D. C.: Mathematical Association of America, 2009.
84. P. Cameron, *Permutation Groups*. Cambridge: Cambridge University Press, 1999.
85. B. I. Plotkin, L. J. Greenglaz, and A. Gvaramija, *Algebraic Structures in Automata and Databases Theory*. Singapore: World Scientific, 1992.
86. R. Goldblatt, *Topoi - The Categorical Analysis of Logic*. Amsterdam, Netherlands: Elsevier Science Publishers B.V., 1984.
87. N. L. Melikadze, "Automata in categories. The decomposition problem," *Cybernetics and Systems Analysis*, vol. 16, no. 4, pp. 469–475, 1980.
88. B. Jacobs and J. Rutten, "A tutorial on (co)algebras and (co)induction," *EATCS Bulletin*, vol. 62, pp. 222–259, 1997. [Online]. Available: <http://eprints.kfupm.edu.sa/21860/>
89. A. Peter and M. N. Paul, "A final coalgebra theorem," in *Category Theory and Computer Science*, ser. Lecture Notes in Computer Science, D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, Eds., vol. 389. Springer, 1989, pp. 357–365.
90. G. Ciobanu and S. Rudeanu, "Equivalent transformations of automata by using behavioural automata," *J. UCS*, vol. 13, no. 11, pp. 1540–1549, 2007.
91. B. Jacobs, "Coalgebraic trace semantics for combined possibilistic and probabilistic systems," *Electron. Notes Theor. Comput. Sci.*, vol. 203, no. 5, pp. 131–152, 2008.
92. D. Varacca, "Probability, nondeterminism and concurrency: Two denotational models for probabilistic computation," University of Aarhus, Tech. Rep., November 2003.
93. E. Badouel and M. T. Tchendji, "Merging hierarchically-structured documents in workflow systems," *Electron. Notes Theor. Comput. Sci.*, vol. 203, no. 5, pp. 3–24, 2008.
94. S. Vickers, *Topology via logic*. New York, NY, USA: Cambridge University Press, 1989.
95. L. S. Moss, "Coalgebraic logic," *Annals of Pure and Applied Logic*, vol. 96, 1999.
96. B. Jacobs, "The temporal logic of coalgebras via galois algebras," *Mathematical Structures in Comp. Sci.*, vol. 12, no. 6, pp. 875–903, 2002.
97. —, *Introduction to Coalgebra. Towards Mathematics of States and Observations*. In preparation, draft electronically available., 2005.
98. A. Kurz, "Coalgebras and their logics," *SIGACT News*, vol. 37, no. 2, pp. 57–77, 2006.
99. I. Hasuo, B. Jacobs, and A. Sokolova, "Generic trace theory," in *International Workshop on Coalgebraic Methods in Computer Science (CMCS 2006)*, volume 164 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, 2006, pp. 47–65.
100. H. Mantel, "A uniform framework for the formal specification and verification of information flow security," Ph.D. dissertation, Universität des Saarlandes, Saarbrücken, Germany, Juli 2003.
101. J. A. Goguen, "Realization is universal," *Mathematical Systems Theory*, vol. 6, no. 4, pp. 359–374, 1972.
102. G. D. Plotkin, "A structural approach to operational semantics," *Journal of Logic and Algebraic Programming*, vol. 60–61, pp. 17–139, 2004.
103. P. Dini and D. Schreckling, "Notes on Abstract Algebra and Logic: Towards their Application to Cell Biology and Security," in *2nd International Conference on Digital Ecosystems and Technologies, IEEE-DEST 2008, 26-29 February, 2008*.
104. J. Lambek and P. J. Scott, *Introduction to higher order categorical logic*. New York, NY, USA: Cambridge University Press, 1986. [Online]. Available: <http://portal.acm.org/citation.cfm?id=7517>
105. S. MacLane, *Categories for the working mathematician*. Springer-Verlag, 1971.
106. S. M. Lane, *Categories for the Working Mathematician (Graduate Texts in Mathematics)*, 2nd ed. Springer, September 1998.
107. G. H. Mealy, "A method for synthesizing sequential circuits," *Bell System Technical Journal*, vol. 34, no. 5, pp. 1045–1079, 1955.
108. D. J. S. Robinson, *A Course in the Theory of Groups*, 2nd ed., ser. Graduate Texts in Mathematics. New York, Berlin, Heidelberg: Springer-Verlag, Aug. 1995, vol. 80.
109. R. Milner, *The Space and Motion of Communicating Agents*. Cambridge: Cambridge University Press, 2009.

Message Diffusion Protocols in Mobile Ad Hoc Networks

Ahmad Al Hanbali¹, Mouhamad Ibrahim², Vilmos Simon³, Endre Varga³ and Iacopo Carreras⁴

¹ Queueing and Performance Analysis group
Eurandom Research Institute Eindhoven

University of Technology
Department of Mathematics and Computer Science
5600 MB Eindhoven
alhanbali@eurandom.tue.nl

² Institut National de Recherche en Informatique et Automatique,

INRIA Sophia Antipolis,
F-06902 Sophia Antipolis, France
Mouhamad.Ibrahim@sophia.inria.fr

³ Department of Telecommunications
Budapest University of Technology and Economics
H-1111, Budapest, Hungary

{svilmos; bacsardi; szabos}@hit.bme.hu

⁴ CREATE-NET

via alla Cascata 56/D
Povo, Trento – 38123, Italy
iacopo.carreras@create-net.org

Abstract. For the last twenty years, mobile communications have experienced an explosive growth. In particular, one area of mobile communication, the Mobile Ad hoc Networks (MANETs), has attracted significant attention due to its multiple applications and its challenging research problems. On the other hand, the nodes mobility in these networks has introduced new challenges for the routing protocols, especially when the mobility induces multiple disconnections in the network. In this chapter, we present an overview of this issue and a detailed discussion of the major factors involved. In particular, we show how messages can be efficiently disseminated in different types of MANETs.

1 Introduction

A Mobile Ad hoc Network (MANET) [8] is a self-configuring network consisting of mobile nodes that are communicating through wireless links. Nodes are free to move and the network transparently supports such movement by dynamically re-configuring itself whenever appropriate. The architecture that nodes form is fully distributed, since they don't assume any centralized network infrastructure to coordinate the communications among them, and each participating node can initiate a peer-to-peer data exchange with any other node through one-hop, or multi-hop paths.

The intrinsic distributed and self-configuring nature of this communication paradigm, combined with the ease and flexibility of deployment of such networks, make MANETs appealing for a wide range of application scenarios including, e.g., emergency situations, sensor networks for environmental monitoring [35] [57], vehicular ad hoc networks [12], and many others [13,20].

The common denominator behind all these application scenarios is the fully distributed nature of the network infrastructure supporting them, together with the support of nodes mobility. In particular, this last characteristic is reflected in a network topology that can change over time, depending on the density and mobility of nodes. With this respect, it is possible to distinguish between more or less dense networks.

The former case corresponds to networks that are connected most of the time, and for which a path almost always exists from a source to a destination. In this case, disconnections represent an exception, rather than the rule, and need to be handled properly, although they do not represent a key requirement during the system design. To this category belong those application scenarios for which the combination of (i) nodes mobility (ii) nodes density and (iii) communication technology guarantees that the network

is partitioned for relatively small time periods.

The latter case is constituted by those scenarios where the devices may be disconnected due to some physical constraint. The most evident case is interplanetary Internet [13], where the planets' orbits drive the presence/absence of line-of-sight and hence the possibility of communications. Another similar case arises when we consider the use of buses and other public transportation means for carrying and disseminating information [12]. Such systems may be used, e.g., to diffuse information about traffic situation, parking availability, special events (conferences, fairs etc.), local advertisement, video-surveillance and similar tasks. In these situations, connectivity cannot be taken for granted, which determines the need to define an architecture able to handle disconnected operations. This category, generally referred as Delay Tolerant Networks (DTNs), is characterized by a disconnected network topology, and nodes use opportunistic forwarding for achieving network-wide communications.

Depending on the specific network topology characteristics – connected or disconnected – the way messages are diffused in the network may vary significantly. Broadcast techniques are more appropriate for the first case, where the density of nodes allows to maximally exploit the intrinsic broadcast nature of the wireless medium for reducing the number of messages being exchanged. Differently, in the case of disconnected networks, the main design goal of forwarding algorithms is shifted from performance to robustness and reliability. In this sense, it is of paramount importance to use redundancy in order to cope with the randomness of network dynamics.

Starting from this differentiation, in this survey we investigate the subtleties of various techniques that have appeared in the literature, and provide a comprehensive survey of the related works. In Sec. 2, we analyse the case of broadcast diffusion techniques, classifying the different methods depending on the side-information they require, and providing a broad overview of the most utilized protocols. Sec. 3 addresses the case of opportunistic communication techniques, describing the main challenges related to the design of forwarding schemes, and detailing the most relevant methods in this area. In Sec. 4, we describe the main performance figures that are typically applied when evaluating diffusion processes in mobile ad hoc networks, and point out some of the most relevant models proposed in the literature. Finally, Sec. 5 concludes the survey by drawing the conclusions of this work, and by pointing out the most promising research directions that are currently being investigated.

2 Broadcast and Dissemination Protocols

Many ad hoc applications rely on the existence of a broadcast medium for the dissemination of some control information. The naive first implementation of this was flooding: every node repeats the message after it is first received. However it was realized very soon that this is very far from optimal, and collisions in the medium can lead to serious congestion and loss of packets. To solve this problem many efficient broadcast techniques were designed that take into account some information about their surroundings, instead of blindly repeating every packet. These algorithms differ in their assumptions about the environment (like assumption of a connected or disconnected network) and in the information available for decision (availability of Global Positioning System (GPS) for example). The central problem of broadcast algorithms is to decide when and who should retransmit messages. Nodes have to forward packets so the message reaches every part of the network; however the performance relies heavily on the set of nodes that do this. When nodes decide whether to retransmit or not, they actually decide if they are part of the forwarding set or not. Too many retransmissions cause collisions and waste the network bandwidth, but choosing the smallest forwarding set is not easy because a global view of the network is not available, and local information gets obsolete very quickly if the velocity of nodes is high. There is also a risk if the number of forwarding nodes is too small, because then the message may not reach every node. In this section we try to give an overview of the existing algorithms and approaches by giving a categorization and showing some of the interesting techniques. There are many possible ways to categorize dissemination protocols. One of the most used is in [67], which we also use as a basis, and extend it where it is needed. Giving a strict, orthogonal categorization, where an algorithm is part of exactly one class is in fact not feasible because there are many hybrid approaches that fall into many categories, and exploit different approaches simultaneously. Instead of this, we provide a usable, but not necessarily rigorous classification of algorithms. This way we can follow the conventions already established in the

field. We try to show what approaches are common for dissemination protocols, and how the existing solutions relate to these approaches. In the following sections (2.1, 2.2 and 2.3) we give three classifications which capture three different aspects of dissemination algorithms. These classes are overlapping, and capture different aspects of the existing algorithms. In 2.1 we show what kind of information can be used to optimize broadcasting.

2.1 Categorization by the used information

The dissemination algorithms use different information about their environment to make their decisions. When one has to choose from the bag of existing algorithms, one has to first investigate if the information needed is available in the target network. Good examples for this are the dissemination methods that use location information. Another example can be the use of some beacon mechanism, where nodes explicitly notify others about their presence. If our network devices cannot acquire the information needed by a broadcast method, then we could not use that algorithm. However, when such information is available, the efficiency of broadcasting can be greatly improved and the use of bandwidth can be reduced.

Simple heuristic based algorithms These algorithms use very limited information about their environment. Usually they do not require periodically updated information about their neighbours, instead they watch the events of their surroundings, like successful transmissions, collisions, or the number of duplicate packets, and try to figure out whether the rebroadcast of data is needed or not. One of the most frequently used environment information is the number of received duplicates of a packet, like in the Counter Based Method [53]. Because these simple algorithms depend on heuristics, they often have some adjustable parameters, which are loosely based on the physical world, and incorporate the intuition of the designer. Determining the optimal value for these parameters is not easy. To solve this problem many of the algorithms in this category have an adaptive version, which tries to figure out the optimal values for the internal parameters. Most of these algorithms are very simple, and they are usually outperformed by the more complex ones. It is also very hard to design good heuristics that actually work in real world scenarios, too. However, in [17] the author shows how simple learning algorithms – like a decision tree learning – are able to mimic the behaviour of complex ones almost perfectly, while using much simpler decision rules than the original ones. This suggests that while good heuristics are very hard to produce “by hand” there are very well-performing heuristics, that can be found by learning algorithms or genetic algorithms (or even by a combination of both).

Neighbour information based algorithms These algorithms use some information about the local topology around the sender. To acquire this, these algorithms need to use periodic HELLO messages that indicate the presence of a node. These beacon messages may contain additional topology information on neighbours of the sender. Some of the algorithms collect knowledge about their immediate neighbours solely, others use k -hop information (where $k = 2$ in most of the cases). In this case, the algorithms know the local topology with higher precision, and so they can coordinate the forwarding of messages much more efficiently. Often these algorithms are sensitive to high node speeds, because their local topology view gets outdated very quickly, so the efficiency of their forwarding policy drops. To overcome this, broadcast algorithms may choose to send topology updates more often, however this can lead to channel congestion.

Location-based algorithms Location-based algorithms use some spatial information to make their decisions. In most of the cases this means that the device should have a Global Positioning System (GPS) to acquire this information. These methods use HELLO messages, just like the neighbour information (section 2.1) algorithms do, but they collect the location of the neighbours too. There are also algorithms that need to know only the distance to their neighbours [39], which may be measured by signal power. In this case, the use of HELLO messages may not be necessary. The location-based algorithms can perform very well (especially when combined with neighbour information), because of their very precise view of the local topology. However, the performance of these algorithms is not well understood when the error of the positioning system cannot be neglected.

2.2 Categorization by strategy

In section 2.1 we showed what kind of information can be used by the different algorithms in the literature. However this information can be processed in different ways. Again, we must emphasize that these categories are not exclusive, and some of the algorithms are put into a separate class just because they are usually discussed together in the literature.

Stochastic These algorithms inhibit some intrinsic random behaviour that does not come from the randomness of the environment. The benefit of this can be the elimination of coupling between the decisions of neighbours, so the mathematical properties of these algorithms may become easier to derive (this is very similar to randomization in statistics). This way, the selection of an appropriate parameter value can be supported by some mathematical results, which is a clear benefit compared to the ad hoc methods sometimes used for setting parameters. The drawback is that in some cases a random behaviour may destroy some information for the neighbouring nodes, as the behaviour of the algorithm is no longer deterministic. Usually the behaviour is adaptable to different situations by adjusting the probabilities of different decisions. Many of the stochastic algorithms are heuristics-based. Some of the algorithms have some Media Access Control that introduces non-deterministic behaviour. While these algorithms could be treated as stochastic algorithms, we prefer to refer to them as deterministic.

Deterministic These algorithms use usually some simple information about their environment (so they usually fall also in the Simple Heuristic Based category, see section 2.1) and behave always in the same way when the environment is the same. One of the simplest examples is the Counter Based Method already discussed in subsection 2.1. Among the deterministic algorithms, those based on graph theory have a special status, and are usually discussed separately. These algorithms also have their own literature [46,68,18,43]; they use more solid mathematical models instead of simple ad hoc heuristics, and usually rely heavily on graph theory results. Basically two types of such algorithms exist, depending on who makes the decision about being a forwarding node or not. Self-pruning methods let every node decide to retransmit or not, while nodes using designation protocols explicitly choose which neighbours should be forwarding nodes.

The self-pruning algorithms collect neighbourhood information from other nodes, and make a local decision whether to forward the message or not [68]. Nodes can decide their forwarding status when the neighbourhood information is updated (on-update) or when the broadcast packet is first received (on-the-fly). The usual goal of these algorithms is to approximate a minimal connected dominating set (MCDS) [38]. A node set is a Connected Dominating Set (CDS) if every node is in the set, or is the neighbour of a node in the set. An MCDS is the smallest of the possible Connected Dominating Sets. An approximation scheme has been adopted to distributed systems in [42,16] primarily for the use in wireless ad hoc networks.

Designation protocols on the other hand try to choose the forwarding nodes among themselves by explicitly designating a node to be a forwarder. Usually a node selects a subset of nodes from his 1-hop neighbours to be forwarding nodes for its 2-hop neighbours. The list of the designated forward nodes is usually sent with the broadcast packet [68]. The goal for neighbour designation protocols is the same as for the self-pruning protocols to approximate a minimal connected dominating set (MCDS). In [68] the authors give a generalization of the idea that incorporates most of the self-pruning and neighbour designation algorithms as special cases.

2.3 Categorization by media access

In every problem that has to be solved over a wireless medium we have to deal with the problem of media access. This is a big difference from wired networks, where much of the details of the network can be abstracted away. Unfortunately wireless networks are quite “hostile”, and irresponsible abstractions of the medium can lead to poorly performing designs. Broadcast algorithms are no different, so we dedicate this subsection to discuss some of the approaches that different algorithms use.

Some methods assume Media Access to be solved by a lower layer. These algorithms usually suppose a MAC mechanism that does not use an RTS/CTS (Ready To Send/Clear To Send) handshake because this can degrade the performance of the broadcasting algorithms. Most of the algorithms use instead a random jitter to minimize collisions with other nodes. This is enough in most of the cases to avoid packet loss. However some of the algorithms use a more elaborate random backoff algorithm. One of the most used approach is called Random Assessment Delay (RAD) which was first introduced in [53]. Algorithms using a RAD mechanism do not retransmit messages immediately, but instead they wait for a random time. During this time they are able to collect more information about their environment. After the RAD expires they can cancel or proceed with the retransmission, according to the events that happened during the RAD. Some algorithms vary the length of the possible RAD period according to local congestion levels. In other cases RAD is used as a prioritization method to make some of the nodes more likely to broadcast first (usually the ones with the most neighbours). We should note that the use of RAD creates non-deterministic behaviour, but we prefer to discuss this under Media Access, and not to treat the resulting algorithms as stochastic. There are also broadcast algorithms, that adjust the signal levels of the network interfaces in a coordinated way, to improve the efficiency of broadcasts. Of course these approaches need specialized equipment at the nodes.

2.4 Survey of existing algorithms

There are many published comparisons of information dissemination methods [40,18,67,2,33], which provide us a quite detailed picture about the existing broadcast approaches. In this section we will try to give a quick overview of the most important methods.

The Counter Based Method, originally introduced in [53], is one of the first controlled broadcast methods, and it is a deterministic, heuristic based algorithm. It is based on a simple observation, that if a duplicate of a packet is received, then the probability of reaching any new node is low. To exploit this idea, the nodes do not immediately transmit on the receipt of a packet, but instead they wait for a random time, which is called Random Assessment Delay (RAD). If a duplicate is received during the RAD a counter is increased. If the counter reaches a threshold before the RAD expires, the node cancels the transmission. The original method has different adaptive versions [2], which try to adapt the length of RAD and the threshold of the duplicate counter to the current network conditions.

Another very simple broadcast method is the Gossiping algorithm which was also introduced in [53]. It is a very simple one: every node broadcasts the heard message with a predefined probability. The optimal probability can be calculated off-line, or can be learned adaptively. Some of these adaptive versions are covered in [15]. While the Counter Based method is a fine example of a simple heuristic based deterministic algorithm, the Gossiping is an example for the simple heuristic based stochastic methods. While it is very easy to implement, it is usually outperformed by other more sophisticated algorithms. Another problem can be, that while the optimal retransmission probability can be calculated off line, it heavily relies on the parameters of the environment. To overcome this limitation there are adaptive versions of the basic methods, like Hypergossiping.

Hypergossiping [37] is one of the most recent algorithms discussed in this document. It is specifically designed for partitioned networks, where nodes are mobile, and partitions join and split from time to time. It is an advanced version of the Gossip algorithm, extended by neighbour information and partition join detection. The algorithm uses a simple adaptive gossiping strategy for in-partition forwarding, but rebroadcasts some of the packets if it detects a join with another partition. The join detection is based on the simple heuristic, that the nodes in the same partition received the same messages recently. Every node maintains a list, called LBR (Last Broadcasts Received), of the recently broadcast messages. They send HELLO messages periodically, to indicate their presence. When a new node is detected, one of the nodes includes its LBR in the next HELLO message. When the other node receives this LBR, it compares with its own LBR. If the overlap between the LBR of two nodes is smaller than a threshold, then the node is considered coming from another partition, so a new message is sent, called BR (Broadcasts Received), which contains the list of messages that the node already received. From this the other node knows that a partition join happened, and rebroadcasts all the messages that were not inside the other nodes BR. After this rebroadcast, dissemination proceeds using adaptive gossiping.

One example for location based protocols is the Optimized Flooding Protocol (OFP) which is a deterministic dissemination algorithm, that uses a geometric approach instead of the usual graph theory solutions. The algorithm tries to cover the 2D space efficiently with R radius circles. We do not detail the algorithm here, mostly because we do not think circles are good approximations of transmission ranges in urban and in-building environments. Details can be found in [62].

The Distance Adaptive Dissemination (DAD) algorithm in [39] uses distance information instead of exact positions. The authors propose a scheme that chooses forward nodes according to their distance, using the signal strength as a measure for distance. The goal of the algorithm is to try to get the outermost neighbours of a node rebroadcast, thus minimizing overlap of transmission ranges. It uses 1-hop neighbour information and records signal levels from the neighbour nodes. The authors propose two variants called DAD-NUM and DAD-PER. DAD-NUM chooses a signal strength S_{thres} so that there are k number of neighbours that have transmitted with a signal strength lower than S_{thres} . On arrival of a new packet, the node checks if the signal strength is greater of S_{thres} or not. If it is smaller then the node rebroadcasts. DAD-PER is very similar, but instead of finding the k farthest nodes it chooses p percent of them.

A fine example of a self-pruning algorithm is the Scalable Broadcast Algorithm (SBA) algorithm (introduced in [45]). It requires 2-hop neighbour information and the last sender ID in the broadcast packet. When a node v receives a broadcast packet from a node u it excludes the neighbours of u , $N(u)$ from the set of its own neighbours $N(v)$. The resulting set $B = N(v) - N(u)$ is the set of the potentially interested nodes. If $|B| > 0$ then the node will start a Random Assessment Delay (RAD). The maximum RAD is calculated by the $\left(\frac{d_v}{d_{max}}\right) \cdot T_{max}$ formula, where $d_v = |N(v)|$ and d_{max} is the degree of the node with the largest degree in $N(v)$, and T_{max} controls the length of the RAD. Nodes choose the time of transmission uniformly from this interval. This ensures that nodes with higher degree often broadcast packets before nodes with fewer neighbours.

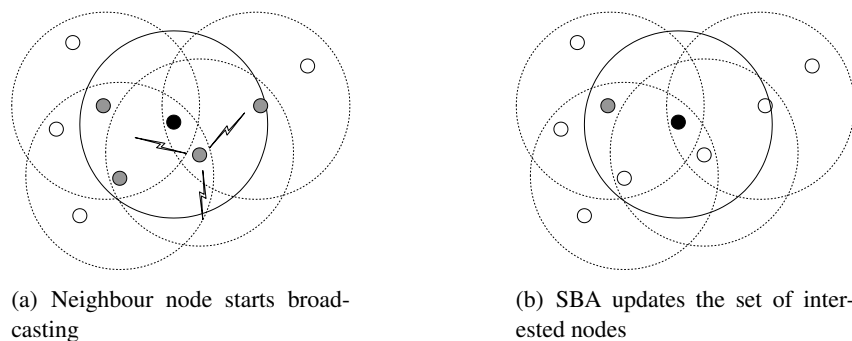


Fig. 1. Demonstration of SBA context update when neighbour node broadcasts

The basic idea of the SBA algorithm was the RAD process, which delays transmission of packets by a random interval. The first time SBA receives a packet, it starts the RAD procedure. During this interval the SBA listens to its neighbours. When one of them starts broadcasting, SBA removes the neighbours of the broadcasting nodes from its own 1-hop neighbour set. This process is demonstrated on figure 1. The length of the RAD depends on the number of the immediate (1-hop) neighbours, so nodes with more neighbours are more likely to transmit first. When the RAD ends, SBA checks if anybody remained, who might be interested in the message. If all of the neighbours are covered by another nodes, SBA cancels transmission, and the algorithm stops.

Multi-Message Scalable Broadcast Algorithm (MMSBA) is a modified version of the original SBA algorithm which is adapted to partitioned networks, and allows the dissemination of multiple messages simultaneously. One of the improvements over SBA is that MMSBA triggers a RAD not only on the first reception of a message, but on any event that changes the local neighbour information. Every time a HELLO message is heard, MMSBA updates the neighbours list. When the number of interested nodes becomes larger than zero because of the detection of a previously unseen node, MMSBA starts the RAD which works exactly the same way as in SBA. There is a little problem though. Every time a node

receives a HELLO message from an unseen node, the algorithm in this form will add him to the list of interested nodes, even if it already had the broadcast message. This problem is not present in the original SBA, as nodes broadcast the message at most once, when it is first received. To overcome this problem, the nodes include the list of messages they have already received in their HELLO packets. This also gives a feedback to MMSBA if a broadcast message was lost during transmission. To support multiple messages, the RAD process also needs to be updated. When a neighbour node broadcasts MMSBA removes from its context the nodes that are interested *only* in that broadcast message. However, nodes that are interested in other messages remain in his list. This mechanism can be imagined as overlapping independent networks, where different messages are disseminated independently using the SBA RAD in the overlapping networks.

The algorithm described in [42], referred to as Wu and Li's algorithm in the literature is a self-pruning algorithm based on a marking process. First, every node is marked as gateway if it has two neighbours that are not connected to each other. To reduce this redundant Connected Dominating Set (CDS), the algorithm uses two rules to prune out unnecessary forward nodes.

Rule 1 A node v can be unmarked if it knows that there is a node u with higher priority that covers all of its neighbours.

Rule 2 A node v can be unmarked if it knows that there are u, w nodes, that are connected, have higher priority than v , and cover all of the neighbours of node v .

The algorithm does not specify how much detail is available to the nodes about their surroundings. In [18] the authors use 2-hop information to compare the performance of the algorithm with other self-pruning methods.

Stojmenovic's method [18,43] is a variant of Wu and Li's algorithm. There are two important improvements over the original algorithm: it uses 1-hop information coupled with position information to implement the marking process and rules 1, 2. The other difference that it also implements a random backoff scheme, similar to SBA. The nodes do not broadcast immediately, but rather wait for a random time. If a node v hears a transmission during this interval from a node u then he removes $N(u)$, the neighbours of u , from its own neighbour set $N(v)$.

Multipoint relaying [64] is a neighbour designation protocol. The designated nodes that relay the messages are called Multipoint Relays (MPR). The nodes send HELLO messages to discover their 2-hop neighbourhood and they try to choose as MAR the node that is able to reach most nodes among the 2-hop neighbours. The algorithm first chooses the nodes from its 2-hop neighbours that are reachable by only one node from the 1-hop neighbours, and assigns MPR status to these 1-hop neighbours. From the remaining set of 1-hop neighbours it chooses the one that covers most of the uncovered 2-hop neighbours. This step is repeated until all of the 2-hop neighbours are covered. This algorithm is also part of the Optimized Link State Routing (OLSR) Internet draft.

The Ad Hoc Broadcast Protocol (AHBP) algorithm (introduced in [46]) is another designation protocol, which is similar to Multipoint relaying but introduces some new ideas. First, designated neighbours (Broadcast Relay Gateway or BRG in AHBP terminology) are not informed in a separate HELLO message, but in the header of the broadcast data. The other difference is that when a node receives a BRG designation, it also checks which neighbours have received the message with the same transaction, and considers these nodes covered when it chooses the next hop BRGs.

A generalization of self-pruning and neighbour designation protocols was introduced first as two general rules in [68] and then specific versions of the rules were used in [18] to make a comparison with other algorithms. The algorithm is referred to as Generic Self-pruning. In its general form the method relies on k -hop neighbourhood and k -hop routing information. The class of algorithms they describe use one of the versions of the so-called Coverage Condition. The most used case is when 2-hop neighbour and 2-hop routing information is used, and the self-pruning made according to the static version of Coverage Condition I⁵: Node v has a non-forwarding status if for any two neighbours u and w a so called *replacement path* exists that connects u and w via several immediate nodes (if any) with either higher priority values than the priority of v or with the visited node status. Generic self-pruning contains many

⁵ Coverage Condition II is a computationally less expensive approximation of Condition I for very simple devices

existing algorithms as special cases of Coverage Condition I or II (both of them are detailed in [68]), for example Lightweight and Efficient Network-Wide Broadcast (LENWB), another neighbour based self-pruning algorithm is in fact a special case of the Coverage Condition from the General Self-Pruning algorithm where the priority of the nodes are given by the number of their neighbours. It uses 2-hop neighbour and 1-hop routing information.

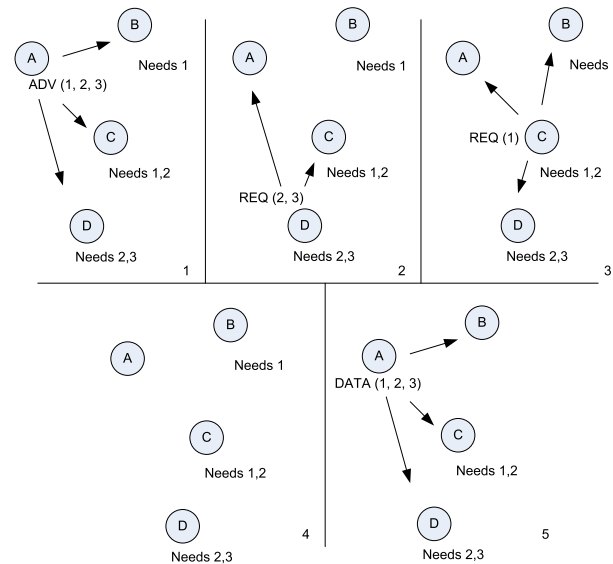


Fig. 2. IOBIO handshake sequence

A quite different approach from the algorithms discussed so far is the IOBIO algorithm[63]. It is a variation of the SPIN [27] dissemination protocol. It uses a simple 3-stage handshake to discover neighbours that are interested in one of the carried messages. The goal of the protocol is to reduce the unnecessary load of neighbouring nodes by duplicate or unneeded data ("spamming"). There are three IOBIO message types that are used by the protocol. The ADV (Advertisement) messages are sent periodically, and they contain the list of messages that the sending node has. Neighbour nodes indicate their interest in the advertised messages by sending a REQ (Request) packet. In response to the REQ, the originator node sends the required DATA packets. The transmission of a REQ after an ADV is not done immediately, but after a random delay. During this delay, the nodes listen to each other, and they only request packets that were not requested before. This process is demonstrated on Figure 2. Node A sends an advertisement indicating that it has the messages 1, 2 and 3. After receiving this ADV, nodes B, C, D start a random delay. At step 2 node D sends a REQ packet, indicating, that he needs messages 2 and 3. At step 3, node C sends a REQ packet. However, he heard the REQ packet of D, so he knows, that message 2 is already requested, and only puts the ID of message 1 in the REQ packet. At step 4, the random delay of B is over, however, message 1 is already requested, so no REQ is sent. At step 5 the wait interval of node A is over, and it broadcasts all requested messages.

2.5 General comparison and discussion

Table 1 summarizes the algorithms discussed in the previous sections. The main aspect of classification of dissemination protocols in dense mobile ad hoc network is the strategy used to effectively propagate information in the system. Simple heuristic based and stochastic methods are usually outperformed by the more sophisticated approaches like neighbour or location based strategies. An important constraint can be the availability of special hardware e.g. GPS devices for location based methods meanwhile most of the neighbour based schemes do not need any additional support. In dense scenarios neighbour based protocols can dramatically decrease redundancy of dissemination at the cost of increasing the overall amount of control messages. Another drawback could be the sensitivity to fast topology changes caused

Algorithm	Multi-message	Forwarding Decision	Control Messages	Special Hardware	Strategy	Sensitivity on Mobility
Counter Based [53]	no	self	none	none	Simple heuristic	Speed and Connectedness
Gossiping [53]	no	self	none	none	Simple heuristic / Stochastic	Speed and Connectedness
Hypergossip [37]	yes	self	LBR and BR	none	Stochastic	Speed
OFP [62]	no	self	HELLO with location	circular radio range; GPS	Geometry based	Speed and Connectedness
DAD [39]	no	self	HELLO	signal strength measurement	Simple heuristic / Location based	Speed and Connectedness
SBA [45]	no	self	2-hop HELLO	none	Neighbor based	High speeds and Connectedness
MMSBA [56]	yes	self	2-hop HELLO with BR	none	Neighbor based	Speed
Wu and Li's algorithm [42]	no	self	2-hop HELLO	none	Neighbor based	Speed and Connectedness
Stojmenovic [18,43]	no	self	2-hop HELLO	GPS	Neighbor based / Location based	High speeds and Connectedness
MPR [64]	no	designated	2-hop HELLO	none	Neighbor based	Speed and Connectedness
AHBP [46]	no	designated	2-hop HELLO	none	Neighbor based	High speeds and Connectedness
Generic [68]	no	self	k-hop HELLO	may use for prioritization	Neighbor based	Speed and Connectedness
LENWB [68]	no	self	2-hop HELLO	none	Neighbor based	Speed and Connectedness
MIOBIO [63]	yes	handshake	BR and REQ	none	Handshake based	Speed

Table 1. Comparison of broadcast algorithm

by high velocity nodes. A different aspect of comparison could be where is the forwarding status decided: at the node itself, or by the previous node. An exception to this is MIOBIO which uses a handshake mechanism which means that a negotiation process is carried out among the interested and forwarding

nodes. Many times it is useful to be able to disseminate messages of different services in parallel, only Hypergossip, MMSBA and MIOBIO provides this feature.

3 Routing Approaches for Intermittently Connected Networks

Intermittently connected networks are a new class of wireless networks that started to emerge recently and to gain extensive efforts from the networking research community. In the literature, these networks are found under different terminology such as sparse or extreme wireless networks, or under another commonly used term disruption/delay tolerant networks (DTNs) [1],[20]. These schemes arise in areas where the network spans over large distances with a low and heterogeneous node density and where the presence of a fixed infrastructure has no great impact on the lack of connectivity of the network. Examples of such networking scenarios include disaster healing and military networks, vehicular networks [54], deep space networks [13], communication between rural zones in toward development countries [11],[26], sensor networks for environmental monitoring [35],[57], and many other networks. Nodes participating to these networks move according to some random or particular mobility model and are generally characterized by scarce resources such as small buffer sizes, limited power and transmission capabilities. Consequently, low throughput, high end-to-end delay and high loss rates describe the default performance features of these networks.

Due to frequent partitions in these networks, instantaneous end-to-end routes do not exist between most of the node pairs, and hence most of the traditional Internet and/or mobile ad hoc routing protocols fail. However, end-to-end routes may exist over time if the nodes can take advantage of their mobility by exchanging and carrying other node messages upon meetings, and by delivering them afterwards to their destinations. The latter concept have gave rise to a novel routing paradigm in these networks called the store-carry-and-forward approach, in which the nodes will basically serve as relays for each others, thus, the term "mobility-assisted routing approach" that is used in conjunction to describe these approaches.

This part will survey and classify various research works that have considered routing schemes for intermittently connected networks. Actually, there are different ways to categorize these approaches. Hereafter, we propose a classification that is based on the degree of knowledge that the nodes have about their future contact opportunities⁶ with other nodes. Specifically, depending on whether these contact opportunities are scheduled, controlled, predicted or opportunistic, these approaches can be grouped into one of the four following families.

3.1 Scheduled-contact based routing

This section surveys the routing approaches that attempt to improve the performance of a sparse network when its dynamics are known in advance such as for instance Low-earth Orbiting satellites (LEO) based networks. In a given network scenario, the most important metrics of interest are the following. The contact times between nodes (their starting times and durations), queue lengths of the nodes, and the network traffic load. The complete knowledge of these three metrics by the routing protocol allows to select optimal routes between the nodes. Despite that the implementation of the complete knowledge in a distributed environment is a very hard task, its evaluation is important as it constitutes the best case scenario compared with other case where only a partial knowledge is available to the routing protocol. On the other side, the approaches that use zero knowledge constitute the worst case scenario.

Jain et al. in [31] use the delay of a link as a cost function, and define the cost of a route to be the sum of its link costs. The authors propose four different techniques that utilize different degrees of knowledge. The first proposal is the Minimum Expected Delay (MED) where only the expectation of the link delay (excluding queueing delay) is known by the routing protocols. The second is the Earliest Delivery (ED) where the instantaneous link delay is available. The third is the Earliest Delivery with Local Queueing (EDLQ) where in addition to the use of the instantaneous delay, the delay at the local queue node is known. The last is the Earliest Delivery with All Queues (EDAQ) where in addition to the link delays, all the delays of the nodes queues are known. All these approaches were evaluated using simulation and

⁶ Two nodes are in contact if they are within transmission range of one another.

compared to the zero knowledge and the complete knowledge cases. Their conclusion is that in networks with plentiful communication opportunities, the need for smart algorithms that require more knowledge is minimal. In situation where resources are limited smarter algorithms (EDLQ and EDAQ) may provide a significant benefits.

3.2 Controlled-contact based routing approaches

In this section, we discuss some routing approaches in DTNs which control the mobility of some dedicated additional mobile nodes in order to improve the network performance by increasing the contact opportunities between participating nodes. The additional mobile nodes can either have fixed predetermined paths conceived in a way to permit them to meet a large number of nodes, or their paths can be adjusted dynamically to meet traffic flows between the nodes. Their main task is to relay packets between the participating nodes by providing a store-carry-forward service. Indeed, by controlling the mobility of the additional nodes, a DTN network administrator would be able to limit the delivery delay and to provide bounds on some other performance metrics of the network. In the literature, several research works have discussed the integration of some special mobile nodes and the design of travel paths of these nodes to meet certain optimization criteria.

Jain et al. in [32] have introduced and modelled an architecture of a sparse network constituted by fixed sensors and extended by mobile nodes called *MULEs* (Mobile Ubiquitous LAN Extensions). *MULEs* move in the network area according to a random mobility model. Their task is to collect data from sensors, buffer it and drop it off later to a set of fixed base stations representing data sinks. The main objective of the architecture is to enhance power saving by allowing sensor nodes to exploit the random mobility of *MULEs* by transmitting their data to these mobile nodes over short range radio links when they pass nearby. To characterize data success ratio and queueing delay at the sensor buffer, the authors introduce a simple stochastic model based on renewal theory and bulk queueing theory. Through simulations, they have also investigated other performance metrics when the system parameters, the number of access points and the number of *MULEs* scale. Their basic observation confirms that an increase in the *MULE* density will improve system performance and leverage resource consumption.

Another controlled-contact routing work that is based on a proactive approach has been introduced in [70]-[71] by Zhao et al.. The approach is termed proactive in the sense that the trajectories of the special mobile nodes, termed as message ferries (MF), are already determined and fixed. Under the assumption of mobility of network nodes, the authors consider two schemes of messages ferries, depending on whether nodes or ferries initiate the proactive movement. In the Node-Initiated Message Ferrying (NIMF) scheme, ferries move around the area according to known routes, collect messages from the nodes and deliver the messages later to their corresponding destinations. Aware of the ferries routes, the mobile nodes can adapt their trajectories to meet the ferries in order to transmit and receive messages. In the Ferry-Initiated Message Ferrying (FIMF), the ferries will move upon service requests to meet the nodes. Specifically, when a node has packets to send or to receive, it generates a service request and transmits it to a chosen ferry using a long range radio. When the ferry receives the request, it adapts its trajectory to meet with the node for packet exchanging using short range radio.

In their former work [70], the focus was on the design of ferry routes to meet certain constraints on throughput requirement and delivery delay in networks with stationary nodes using a single ferry. By formulating the problem as two optimization sub-problems, they developed algorithms to design the ferry route. In a recent work [72], they considered the case of multiple ferries with the possibility of interaction between the ferries. The addition of multiple ferries has the advantages of improving the system performance and robustness to ferry failure at the cost of increasing the complexity of the problem. Based on several assumptions regarding whether the ferries follow the same or different routes and whether they interact with each others, they investigated four different route design algorithms that attempt to meet the traffic demand and minimize the delivery delay. Simulation results showed that when the traffic load is low, the impact of increasing the number of ferries on the delivery delay is minor. However, for high traffic load scenarios, the impact is significant.

In [14], the authors propose an algorithm called MV routing which, on one side, exploits the movement patterns of participating nodes, that is the *meeting* and *visit* behaviours, and on the other side, attempts to control the motions of some additional external nodes. Their aim is to improve network efficiency in terms of bandwidth and latency of message delivery. The algorithm is seen as being constituted by two separate mechanisms. Building on their previous work in [19], the first mechanism is a slightly modified variant of the Drop-Least-Encountered technique that is used as a routing strategy instead of a buffer management technique as it has been used in [19]. The second mechanism of the algorithm consists in adapting dynamically the movement paths of some additional nodes to meet the traffic demands in the network while optimizing some performance criterion. Travel path adjustment is carried out through multi-objective control algorithms with the objective of optimizing simultaneously several metrics related to bandwidth and delay. Simulation results demonstrate that exploiting node mobility patterns in conjunction with multi-objective control for autonomous nodes have the most significant performance improvements.

3.3 Predicted-contact based routing approaches

Predicted routing techniques attempt to take advantage of certain knowledge concerning the mobility patterns or some repeating behavioural patterns. Based on an estimation of that knowledge, a node will decide on whether to forward the packet or to keep it and wait for a better chance. Basically, each node is assigned a set of metrics representing its likelihood to deliver packets to a given destination node. When a node holding a packet meets another node with a better metric to the destination, it passes the packet to it, hence increasing the packet likelihood of being delivered to its corresponding destination. According to the nature of knowledge, we propose to reclassify the algorithms falling under this category as based on mobility-pattern or based on history.

Mobility-pattern based approaches Approaches falling under this section attempt to take advantage of common behaviours of node mobility patterns in the network in order to derive decisions on packet forwarding. In fact, by letting the nodes learn the mobility pattern characteristics in the network, efficient packet forwarding decisions can be taken. Two main issues are related to these approaches. The first issue concerns the definition and the characterization of the node mobility pattern where several ways can exist to characterize and acquire such a pattern. For instance, the appearance of stable node clusters in the network, or the acquisition of statistical information related to meeting times or to the visit frequencies of nodes to a given set of locations are examples of mobility patterns that can be exploited by the nodes. The second issue is related to the way through which a node can learn and acquire its own pattern as well as those of other nodes. In particular, the presence of some external signals to the nodes such as GPS coordinates or some fixed beacons help greatly the nodes to acquire easily the mobility patterns in the network. Alternatively, nodes can also learn their own mobility patterns without any external signal by relying only on previous observations and measurements, or by exchanging pattern information with other nodes. Several routing works in DTNs that use mobility patterns to derive forwarding decisions have appeared in the literature.

In [55], the authors develop a routing algorithm that exploits the presence of concentration points (CPs) of high node density in the network to optimize forwarding decisions. The appearance of CPs is seen as the result of a general mobility model where nodes will have a high concentration inside these CPs with random movements over time between these islands of connectivity. The basic idea of their algorithm is to make use of the neighbour set evolution of each node without using any external signals. Specifically, nodes that belong to a given concentration point will collaborate between them to assign a label to their CP. Nodes will learn the labels of other nodes when they move in the network between the different CPs. Using the knowledge of the CP graph, and the positions of the source and destination nodes in the graph, the message is forwarded from the source to its destination through a sequence of CPs using the shortest path between the respective CPs. Even though the algorithm performs well, the need to manage and update the labels introduces some complexity in the algorithm mechanism.

In another work [41], the authors introduce a virtual-location routing scheme which makes use of the frequency of visit of nodes to a discrete set of locations in the network area in order to decide on packet forwarding. Specifically, they define a virtual Euclidean space, termed as *MobySpace*, where the dimension degree and the type of the coordinate space depend on the mobility pattern of the nodes. For instance, for a network with L possible node locations, the *MobySpace* is an n -dimensional space where $n = |L|$. Each node is represented in that space by a virtual coordinate termed as *MobyPoint*. A source node X with a message to send at time t will forward its message to a node Y among the set of its neighbours $W_X(t)$ for which the Euclidean distance to the destination is the smallest. Observe that the *MobyPoint* of a node is not related to its physical GPS coordinate. The acquisition of the visit frequencies of the nodes to the location set is obtained by computing the respective fraction of time of being in a given location.

Another subclass of mobility-pattern based approach consists in exploiting the underlying structure of social aspects of the network, whether in terms of contact patterns as well as set of interests, in order to derive decisions on packet forwarding. Actually, accounting for the social interactions and the social structure of the network to which the mobile users belong was proved to significantly influence the routing performance of the network. Various groups have recently started investigating the impact of social aspects on forwarding protocol design and routing performance.

In [28], the community structure behind the social interactions has been studied in order to improve the forwarding algorithms in the network. The authors showed that there exists a limited set of nodes, called *hubs*, which play a central role in the diffusion of information. Being aware of the community structure, the authors showed that an extremely efficient trade-off between resources and performance can be achieved.

In [48], the impact of different social-based forwarding schemes were evaluated on real world mobility patterns obtained from Bluetooth proximity measures. The authors showed that incorporating a friend/stranger classification in the forwarding policies can be beneficial in different application scenarios.

History based approaches History based approaches are developed mainly for heterogeneous mobility movements. They rely on the observation that the future node movements can be effectively predicted based on repeating behavioural patterns. For instance, if a node had visited a location at some point in time, it would probably visit that location in another future time. Actually, if at any point in time a node can move randomly over the network area, an estimate based on previous contacts is of no help to decide on packet forwarding. However, if the mobility process has some locality, then last encounter times with other nodes can be associated with some weights that can be ranked based on their likelihood to deliver the messages to the corresponding destinations. The following works illustrate the working mechanisms of some of these approaches.

One of the pioneer work that considered history-based routing in sparse mobile networks is the work of Davis et al. in [19]. The objective of their work is to study the impact of different buffer management techniques on an extended variant of the epidemic protocol [61] on nodes with limited buffer size. Even though their work is not related to routing, the way by which the packets are sorted upon a contact influences implicitly the performance of the routing protocol. More precisely, when two nodes meet, they will first transfer the packets destined to each other, then they will exchange the lists of their remaining stored packets. The combined list of remaining packets is next sorted according to the used buffer management strategy, and each node will request the packets it does not have among the top K sorted packets. The authors have explored four different buffer management techniques, among them the *Drop-Least-Encountered* (DLE) technique which makes use of previous contacts with other nodes to decide on packet ranking. Basically, nodes using the DLE technique keep a vector indexed by addresses of other nodes where each entry estimates the likelihood of meeting the corresponding node. At each time step, a given node A updates its likelihood meeting values for every other node C with respect to the co-located node B according to the temporal difference rule (see [60]). If node A meets B , it is likely that A meets B again in the future, and hence A is a good candidate for passing the packets to B . Thus, node A should increase its likelihood for node B . If B has a high encounter for node C , then A should increase its

likelihood of meeting C by a factor proportional to the likelihood of meeting between B and C . Last, if at a given time step, node A did not meet any other node, the different likelihood values decrease in a constant rate.

In [35], the authors propose a wireless peer-to-peer networking architecture, called ZebraNet system, which is designed to support wildlife tracking for biology research. The network is basically a mobile sensor network, where animals equipped with tracking collars act as mobile nodes which cooperate between them in a peer-to-peer fashion to deliver collected data back to researchers. Researcher base stations, mounted on cars, are moving around sporadically to collect logged data. The design goal is to use the least energy, storage and other resources necessary to maintain a reliable system with a very high data delivery success rate. To attain these objectives, they propose the use of a history-based protocol to handle packet transfer between neighbour peer nodes. More precisely, each node will be assigned a hierarchical level based on its past successes of transferring data to the base station. The higher the level of the node, the higher the probability that this node is within range of base station or within range of some other nodes near the base station. Therefore, it has a high likelihood of relaying the data back to the base station either directly or indirectly through minimal number of other nodes. The mechanism works as follows: each time a node scans for peer neighbours, it requests the hierarchy level of all of its neighbours. Collected data is then sent to the neighbour with the highest hierarchy level. Whenever a node comes within range of the base station, its hierarchy level is increased while it is decreased over time at a given rate when it is out-of-range.

Jones et al. in [34] propose a variant of the MED approach of [31] called Minimum Estimated Expected Delay (MEED). Alternatively to the MED approach where the expected delay of a link is computed using the future contact schedule, MEED uses an estimation of the observed contact history. The estimator implements a sliding history window with an adjustable size. To minimize the overhead induced by the frequent updates of the estimated link delay, the authors propose to filter update samples having small difference with the actual information in the network. Through simulations, MEED has shown to overcome the performance of MED as it is more responsive to network changes, and its performance approaches that of the epidemic protocol. However, the algorithm lacks the presence of an adjustment mechanism of its window size.

The authors in [44] propose PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity), a single copy history-based routing algorithm for DTNs. Similarly to [19], each node in PROPHET will attempt to estimate a delivery predictability vector containing an entry for each other node. For a given node X , the entry $P(X, Y) \in [0, 1]$ will represent the probability of node X to deliver a message to a given node, for instance node Y in this case. The entries of the predictability vectors will be used to decide on packet forwarding. Specifically, when two nodes meet, a message is forwarded to the other node if the delivery predictability for the destination of the message is higher at the other node. In addition to the predictability vector, a summary vector of stored packets will be also exchanged upon contact. The information in the summary vector is used to decide on which messages to request from the other node. The entry update process occurs upon each contact and works as follows. Nodes that are often within mutual ranges have a high delivery predictability for each other, and hence they will increase their corresponding delivery predictability entries. Alternatively, nodes that rarely meet are less likely to be good forwarders of messages to each other, and hence they will reduce their corresponding delivery predictability entries.

3.4 Opportunistic-contact based routing approaches

Opportunistic based approaches are generally characterized by random contacts between participating nodes followed by potential pair-wise exchanges of data. Given that connectivity, and consequently, data exchanges are subject to the characteristics of the mobility model which are in general unpredicted, these approaches rely on multi-copy schemes to speed up data dissemination within the network. In the following, we subdivide these approaches into epidemic-based approaches and coding based approaches.

Epidemic based approaches Epidemic based approaches imitate the spread of contagious disease in a biological environment. Similarly to the way an infected individual passes on a virus to those who come into contact, each node in an epidemic-based system will spread copies of packets it has received to other susceptible nodes. The number of copies that an infected node is allowed to make, termed as the fan-out of the dissemination, and the maximum number of hops that a packet is allowed to travel between the source and the destination nodes, represented by a hop count field in the packet, define the epidemic variant of the algorithm. These two parameters can be tuned to trade delay for resource consumption. Clearly, by allowing the packet to spread throughout the mobile nodes, the delay until one of the copies reaches the destination can be significantly reduced. However, this comes at the cost of introducing a large overhead in terms of bandwidth, buffer space and energy consumption. Several variants of epidemic-based approaches have been proposed and their performance in terms of delay and resource consumption have been evaluated.

One of the pioneer work in this domain is the epidemic routing protocol of Vahdat and Becker [61]. The protocol is basically a flooding mechanism accommodated for mobile wireless networks. It relies on pair-wise exchanges of messages between nodes as they get in contact with each other to eventually deliver the messages to their destinations. Each node manages a buffer containing messages that have been generated at the current node as well as messages that has been generated by other nodes and relayed to this node. An index of the stored messages called a summary vector is kept by each node. When two nodes meet, they will exchange their summary vectors. After this exchange, each node can determine then if the other node has some messages that was previously unseen by it. In this case, it will request the missing messages from the other node. To limit the resource utilization of the protocol, the authors propose to use a hop count field at each message that specifies the total number of epidemic exchanges that a particular message may be subject to. They showed that by appropriately choosing the maximum hop count, delivery rates can still be kept high while limiting resource utilization.

In [24], Grossglauser and Tse introduce a one copy two-hop relay protocol. Basically, at any time, they will be one copy of the packet in the network, however, the copy can make at most two hops between the source node and the destination node. Their packet dissemination algorithm can be seen as an epidemic-like protocol with a fan-out of one and a hop count of two. The key goal of their work is to show that the capacity of a mobile network can scale with the number of nodes by exploiting the mobility of these nodes through a two-hop relay protocol.

Building on [61] and [24], several research works have appeared subsequently which proposed analytical models to evaluate the performance of these protocols. In [23] Groenevelt et al. introduce a multicopy two-hop relay protocol (MTR), a variant of the two-hop relay protocol. In MTR, the source forwards a copy of the packet to any other relay node that it encounters. Relay nodes are only allowed to forward the packets they carry to their destinations. By modelling the successive meeting times between any pair of mobile nodes by Poisson processes, the authors characterize the distribution of the delivery delay and that of the total number of copies generated until the packet delivery. This work was extended in [5] by Al Hanbali et al. under the assumption of limited lifetime of the packets, and in [4] under the assumption of general distribution of inter-meeting times. Zhang et al. in [69] extend the work in [23] by evaluating several variations of the epidemic protocol and some infection-recovery schemes. Inspired by [23], the authors of [29] consider a sparse mobile ad hoc network equipped by throwboxes. Throwboxes are small and inexpensive wireless devices that act as fixed relays and that are deployed to increase contact opportunities among the nodes. By modelling the meeting times between a mobile and a throwbox as a Poisson process, the authors characterize the delivery delay and the total number of copies generated under the MTR and the epidemic protocol for the cases where throwboxes are fully disconnected or mesh connected.

A biological acquisition system termed as the shared wireless infostation model (SWIM) has been introduced in [57] as a way of routing collected measurement traces between a set of sensors attached to whales and a set of fixed infostations acting as collecting nodes. Infostations act as base stations which connect the users to the network. Mobile nodes represented by the tagged whales move randomly within the area and connect to the infostations when they are within range to offload their data. When two tagged whales meet, an epidemic exchange mechanism takes place in order to accelerate the delivery

of the packets at the cost of increasing the storage space at the nodes. Through simulations, the authors showed that sharing the data among the whales as well as increasing the number of SWIM stations reduce significantly the end-to-end delay. The positions of infostations as well as the mobility of whales greatly affect the system performance.

Spyropoulos et al. introduce a new routing algorithm for sparse networks in [59], termed Spray and Wait algorithm. The algorithm disseminates a number of copies of the packet to other nodes in the network, and then waits until one of these copies meets the destination. It consists of two phases. In the first phase, the source node will generate a total of L copies of the message it holds, then spreads these copies to other nodes for delivery to the destination node. The spreading process works as follows. When an active node holding $n > 1$ copies meets another node, it hands off to it $F(n)$ copies and keeps for itself the remaining $n - F(n)$ copies and so forth until a copy of the message reaches the destination. F is the function that defines the spreading process. For instance, for binary spray and wait, $F(n) = \frac{n}{2}$. In the second phase, the wait phase, if the destination is not found among the L copy-carrying nodes, then these latter nodes will perform direct transmissions to the destination node. Using simulations, the authors show that this technique can achieve a trade-off between efficient packet delivery and low overhead if the parameters are carefully designed.

Coding based approaches The approaches in Section 3.4 are primarily based on packet flooding in order to improve the efficiency of packet delivery. Unfortunately, these improvements come at the expense of introducing large overhead in the network due to redundant packet transmissions. The approaches presented in this section alleviates the effect of flooding through the use of smarter redundant algorithms that are based on coding theory. In the following, we consider two main coding algorithms that appeared in the literature and which have shown their suitability to the opportunistic contact networks, namely the erasure coding and the network coding.

In the erasure coding scheme, upon receiving a packet of size m , the source produces n data blocks of size $l < m$. The coding algorithm composes these blocks in a such way to allow the destination to retrieve the original message on receiving any subset of these blocks [49]. More precisely, the transmission of the packet is completed when the destination receives the k th block, regardless of the identity of the $k \approx m/l < n$ blocks it has received. The blocks are forwarded to the destination through the relay nodes according to store-carry-and-forward approach. The performance analysis of this approach in opportunistic contact network has shown to improve significantly the worst case delay with fixed amount of overhead [4,65]. Further, in [30] it has been shown that erasure coding improve the probability of packet delivery in DTNs with transmissions failures.

In the network coding scheme, instead of simply forwarding the packets, nodes may transmit packets with linear combinations of previously received ones. For example, consider the three nodes case where nodes A and C want to exchange packets via the intermediate node B . A (resp. C) sends a packet a (resp. c) to B , which in turn broadcasts $a \text{ xor } c$ packet to A and C . Both A and C can recover the packet of interest, while the number of transmissions is reduced. In [66], different aspects of the operability of network coding with limited storage resources have been discussed and different techniques have been proposed. The main result is that network coding benefits more from node mobility and performs well in scenarios of high packet drop rate where simple flooding approaches fail.

3.5 General comparison and discussion

Table 2 compares the various proposals that have been addressed in Section 3 by summarizing the main distinguishable features of each one. Our comparison is based on four features. The first feature defines the degree of knowledge that the nodes have about their future contact opportunities. Future contact opportunities are identified as being scheduled, controlled, predicted or opportunistic. The second feature lists the key relevant performance metrics that each proposal attempts to optimize. For instance, these metrics range from increasing the packet delivery ratio to reducing the end-to-end delivery delay, energy consumption and/or buffer occupancy of the nodes. The third and fourth features list the characteristics of the mobility patterns of the network nodes and the dedicated special nodes, whenever employed. Precisely, nodes of the network could be stationary where in this case they are the special nodes that move

Proposal	Contact opportunities	Metric to optimize	Mobility pattern of network nodes	Mobility pattern of special nodes
Jain et al. protocol [31]	Scheduled	Delay	Random	–
MULE protocol [32]	Controlled	Power usage, Buffer overhead	Stationary	Random
MF protocol [70]	Controlled	Delivery rate, Power usage	Stationary	Predetermined paths
Extended MF protocol [71]	Controlled	Delivery rate, Power usage	Random, Stationary	Predetermined, Dynamic paths
MV protocol [14]	Controlled	Delivery rate, Delay	Meeting and Visit dependant	Metric dependant paths
Island hopping protocol [55]	Predicted	Delay, Transmission overhead	Heterogeneous mobility	–
MobySpace protocol [41]	Predicted	Delivery rate, Power usage	Location dependant	–
DLE protocol [19]	Predicted	Buffer usage	Heterogeneous mobility	–
ZebraNet [35]	Predicted	Delivery rate, Power, Storage	Heterogeneous mobility	–
MEED protocol [34]	Predicted	Delay, Transmission overhead	Random	–
PROPHET [44]	Predicted	Delivery rate, Power usage	Heterogeneous mobility	–
Epidemic protocol [61]	Opportunistic	Delivery ratio, Delay	Random	–
Two-hop protocol [24]	Opportunistic	Network capacity	Random	–
MTR protocol [23]	Opportunistic	Delay, Transmission overhead	Random	–
SWIM protocol [57]	Opportunistic	Delivery rate, Delay	Random, Stationary	–
Spray and Wait [59]	Opportunistic	Delivery rate, Power usage	Random	–
Erasure coding [65]	Opportunistic	Delivery rate	Random	–
Network coding [66]	Opportunistic	Delivery rate	Random	–

Table 2. Summary of the routing approaches in DTNs and their main properties.

around according to some predetermined or dynamic paths to assist in packet routing to the fixed nodes. Alternatively, node mobility pattern could be either random, where there is no means to predict the potential future contacts of a node, or heterogeneous with some location dependency or some correlated meeting among the nodes. Observe that the properties we have listed are not exhaustive and other properties can be included in addition. For instance, the complexity of the proposal in terms of implementation or computation, or the requirement to exchange some control information can also be considered. However, we restricted the comparison to the previous four features, which we think are the most relevant according to the classification that we made before.

4 Modelling approaches

In the absence of predictable mobility and network topology, the notion of “route” for a message to follow loses its significance, and it becomes imperative to employ some kind of “epidemic spreading” mechanisms [36,69]. Depending on the application scenario considered, such epidemic spreading can occur over large periods of time, as in the case of sparse networks where the delivery of messages is obtained from the physical mobility of nodes, or shorter ones, where the dense nature of the network allows to exploit the use of broadcasting algorithms. In both cases, it is of paramount importance to use redundancy in order to cope with the randomness of network dynamic. At the same time, forwarding operations rely on the ability of a node to keep (even for a rather long time) a message in its internal memory. This is justified by the fact that a node may be doomed to remain isolated for a long time, but should still be able to forward the messages it received. In this sense, the redundancy encompassed by the algorithm stresses the existence of a performance/robustness vs. storage/energy consumption tradeoff. Indeed, the larger the number of copies of a message in the system, (i) the faster it reaches its destination (ii) the more it is robust with respect to the nodes mobility and node/link failures. On the other hand, in order to have more copies of the same message travelling in the network at the same time, a larger amount of network resources has to be exploited. Resources are intended in terms of both (i) storage, necessary to keep the message in the nodes’ memory for a longer time (ii) energy consumption, in that a larger number of transmissions of the same message is needed.

From these considerations emerges how the performance of message diffusion in MANETs is always a trade off between different requirements, and single aspects can not be considered in isolation. As an example, end-to-end delay should always be considered as a function of the resources, e.g., storage, allocated to run a specific forwarding or broadcasting algorithm.

It also clear the need to perform, where possible, an accurate modelling of the system and of the various processes occurring in the network, able to efficiently account for the various system parameters and to provide useful insights into the design space of such systems. This need is confirmed by the many models and modelling techniques appeared in the literature over the past years.

The traditional store-and-forward routing protocols, which require the existence of a connected path between a source and a destination, do not achieve good performance in intermittently connected ad hoc networks. A solution for this problem is to exploit the mobility of nodes present in the network. Such an approach is known as store-carry-and-forward and it has been proposed in the pioneering paper of Grossglauser and Tse [24].

The important aspects in the store-carry-and-forward solutions are the so-called contact opportunity and inter-contact time between nodes that mainly depend on the mobility of the nodes. In the following we will first introduce the performance metrics of interest before surveying the performance evaluation tools used in the literature. We should emphasize that most of the performance models developed in the literature focus on the opportunistic networks in Section 3. The key performance metrics in intermittently-connected networks are the following: (i) the network throughput known also network capacity, (ii) the delivery rate defined as the percentage of packets that successfully reach the destination, (iii) the packet delay denoted as the time that a packet requires to reach the destination, (iv) the energy consumption of the network in order to deliver a packet to its destination. The latter metric is especially important for the multicopy relay protocols that belong to the opportunistic class in Section 3.

A significant research work spawned exploring the trade-offs between the capacity and the delay of the two-hop relay protocol and other similar schemes, especially their scaling laws when the number of nodes is large [21,22,24,25,51]. It is important to mention that most of these studies assume a uniform spatial distribution of nodes, which is the case, for example, when nodes perform a symmetrical Random-Walk over the region of interest [22], or when nodes move according to the Random Direction model [50]. Using a queueing analysis the authors in [3] prove that the uniform mobility models achieves the minimal relay throughput as compared with non-uniform models such as the Random-Waypoint model [9]. On the other hand, the authors in [23] show that the distribution of the inter-meeting times between any mobile nodes pair is approximately an exponential distribution. This finding has been noticed for a number of mobility models (Random Walk, Random Direction, Random Waypoint) in the case when the node transmission range is small with respect to the area where the nodes move. Exploiting this property, a

batch of Markovian models of the number packet copies has been proposed recently in the literature to evaluate the delay and the energy consumption of a class of multicopy relay protocols, e.g. MTR, Epidemic Routing, for both the cases of finite and infinite number of nodes [5,4,23,29].

Another tool that was used to evaluate the performance of multicopy relay protocols is the so-called fluid approach also know as the mean field approach. In disconnected mobile networks the fluid quantity represents the mean number of packet’s copies in the network. The dynamics of these quantities in time can be written as a set of ordinary differential equations (ODEs). Using this tool Small and Haas in [57] provide a model, to evaluate the performance of disconnected mobile networks embedded in an infostation network architecture. They consider the case where the Epidemic Routing protocol is used to relay data from the mobile nodes to the infostations. An infostation can be seen as a wireless access port to the Internet or to some private networks. Zhang et al. in [52] extend the work in [57] and showed that the ODEs can be derived as limits of Markovian models under a natural scaling as the number of nodes increases. Moreover, they studied variations of the Epidemic Routing protocol, including probabilistic routing and recovery infection schemes.

Once the performance metrics of interest are computed, e.g. the expected delay and the expected energy consumed, one can construct a number of optimization problems. To this end, certain metrics should be first parametrized such as the maximal number of packet transmissions or the maximal number of packet copies in the case where packet’s copies have limited lifetime. Based on this idea the authors in [59,58] proposed to limit the maximal number of packet’s copies of forwarding protocols using token based solution. Building on these studies Neglia and Zhang in [52] identify the best policy that a node should employ in order to minimize the linear cost function of the expected delay and the expected energy consumption. This is done with the help of the Dynamic Programming theory with a centralized controller.

In the case of dense ad hoc networks most of the modelling approaches of epidemics can not be applied. The most important limitation is that mobility can not be modelled by exponential intermeeting times because of the significant probability of having a node already in range. Also meetings can not be assumed pairwise any more because small connected islands can be formed time to time. Because of connectivity the dissemination delays inside islands are much lower than in DTNs, therefore if the movement speed of nodes is small (pedestrian) the underlying connection graph can be assumed to be fixed during the dissemination in the island. The connections will change significantly only in the timescale of island intermeeting times. This naturally leads to several graph-theory based approaches. One significant use is to model the connection between nodes with Unit Disk Graphs (UDG) [10]. A UDG is constructed by placing unit radius disks on the plane associating a vertex to each circle and connecting the vertices if the corresponding disks overlap. It was shown by Breu and Kirkpatrick in [10] that the problem of deciding that a given graph is a Unit Disk Graph is NP hard.

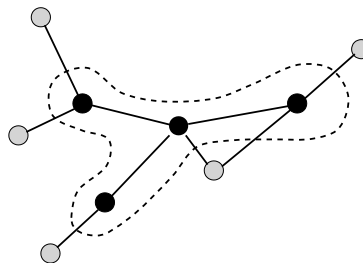


Fig. 3. A Minimum Connected Dominating Set in a graph

Random UDGs share many of the properties of Bernoulli random graphs [6,7]. The most important property of UDGs however is that many hard optimization problems on graphs can be approximated effectively on UDGs [47]. These problems include the approximation of a Maximum Independent Set, Minimum Dominating Set and Minimum Connected Dominating Set [38,42,16] (MCDS) that has very important applications for multi-hop broadcasting. A set of vertices is called a Connected Dominating

Set if every vertex is in the set or has a neighbor in the set and the vertices of the set form a connected subgraph. An MCDS is the smallest of the possible Connected Dominating Sets. Figure 3 shows an example of an MCDS. The cardinality of an MCDS is a lower bound on the number of transmissions that are needed to disseminate a message in a connected island, therefore many algorithms try to approximate an MCDS in a distributed way.

5 Conclusion

In this survey, we have investigated several techniques for packet dissemination in mobile ad hoc networks. By referring to the type of applications which these techniques are designed for, we have categorized them into two generic classes where the first class includes reliable dissemination mechanisms using broadcast as a central means for packet delivery while the second class includes techniques that are designed for networks tolerating high delivery latency where store-carry-and-forward paradigm is the commonly used mechanism. For each class, we have reviewed a large part of recent research works that have appeared and proposed further categorizations of the different techniques according to some distinguishing features.

References

1. Delay tolerant networks (DTN) research group. <http://www.dtnrg.org>.
2. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. In *Proc. of the ICDCS*, Washington, DC, USA, 2001. IEEE Computer Society.
3. A. Al Hanbali, A. A. Kherani, R. Groenovel, P. Nain, and E. Altman. Impact of mobility on the performance of relaying in ad hoc networks- extended version. *Elsevier Computer Networks*, 51(14):4112–4130, Oct. 2007.
4. A. Al Hanbali, A. A. Kherani, and P. Nain. Simple models for the performance evaluation of a class of two-hop relay protocols. In *Proc. of IFIP Networking*, Atlanta, GA, USA, May 2007.
5. A. Al Hanbali, P. Nain, and E. Altman. Performance of ad hoc networks with two-hop relay routing and limited packet lifetime. In *Proc. of IEEE/ACM ValueTools*, Pisa, Italy, Oct. 2006.
6. M. J. Appela and R. P. Russob. The connectivity of a graph on uniform points on $[0, 1]^d$. *Statistics and Probability letters*, 60:351–357, 2002.
7. S. R. Ashish Goel and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. *ACM Symposium on Theory of Computing*, 2004.
8. S. Basagni, M. Conti, S. Giordano, and I. Stojmenovi. *Mobile Ad Hoc Networking*. IEEE Press John Wiley, 2004.
9. C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks*, 10(5):555–567, Sept. 2004.
10. H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry. Theory and Applications*, 9(1-2):3–24, 1998.
11. E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedeveschi, J. Pal, R. Patra, S. Surana, and K. Fall. The case for technology in developing regions. *IEEE Computer*, 38:pp. 25–38, May 2005.
12. J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. of IEEE INFOCOM*, April 2006.
13. S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41:pp. 128–136, June 2003.
14. B. Burns, O. Brock, and B. Levine. MV routing and capacity building in disruption tolerant networks. In *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005.
15. J. Carle, J. Cartigny, and D. Simplot. Stochastic flooding broadcast protocols in mobile wireless networks. Technical Report 2002-03, LIFL Univ. Lille 1, France, May 2002.
16. X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.
17. M. D. Colagrosso. Intelligent broadcasting in mobile ad hoc networks: Three classes of adaptive protocols. *EURASIP Journal on Wireless Communications and Networking*, 2007. Article ID 10216.
18. F. Dai and J. Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):1027–1040, 2004.
19. J. Davis, A. Fagg, and B. Levine. Wearable computers as packet transport mechanisms in highly partitioned ad hoc networks. In *Proc. of 5 IEEE Intl. Symp. on Wearable Computers*, Zurich, Switzerland, 2001.
20. K. Fall. A delay tolerant network architecture for challenged internets. In *Proc. of ACM Sigcomm*, Karlsruhe, Germany, Aug. 2003.
21. R. M. G. Sharma and N. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. In *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.
22. A. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *Proc. of IEEE INFOCOM*, Hong Kong, Mar. 2004.

23. R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1-4):210–228, Oct. 2005.
24. M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *ACM/IEEE Transactions on Networking*, 10(4):477–486, Aug. 2002.
25. P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, Mar. 2000.
26. A. A. Hasson, D. R. Fletcher, and D. A. Pentland. A road to universal broadband connectivity. In *Proc. of Development by Design*, Dec. 2002.
27. W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. MOBICOM*, Seattle, 1999.
28. P. Hui and J. Crowcroft. Bubble rap: Forwarding in small world dtns in ever decreasing circles. Technical Report UCAM-CL-TR-684, Univ. of Cambridge, Computer Laboratory, May 2007.
29. M. Ibrahim, A. Al Hanbali, and P. Nain. Delay and resource analysis in manets in presence of throwboxes. *Performance Evaluation*, 64(9-12):933–947, Oct. 2007.
30. S. Jain, M. Demmer, R. Patra, and K. Fall. Using redundancy to cope with failures in a delay tolerant network. In *Proc. of ACM Sigcomm*, Philadelphia, PA, USA, Aug. 2005.
31. S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant networking. In *Proc. of ACM Sigcomm*, Portland, OR, USA, Aug. 2004.
32. S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy-efficient data collection in sensor networks. In *IEEE WiOpt*, 2004.
33. T. Jin, K. Yunjung, and M. G. Yi. Efficient flooding in ad hoc networks: a comparative performance study, 2003.
34. E. P. Jones, L. Li, and P. A. Ward. Practical routing in delay-tolerant networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.
35. P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *Proc. of ASPLOS-X*, San Jose, CA, Oct. 2002.
36. A. Khelil, C. Becker, J. Tian, and K. Rothermel. An epidemic model for information diffusion in manets. In *Proc. of MSWiM*, pages 54–60, New York, NY, USA, 2002. ACM Press.
37. A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel. Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Netw.*, 5(5):531–546, 2007.
38. S. Khuller and S. Guha. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, 1996.
39. S. Krishnamurthy, X. Chen, and M. Faloutsos. Distance adaptive (dad) broadcasting for ad hoc networks. In *Proc. of MILCOM*, Oct. 2002.
40. T. O.-N. Larsen, T. H. Clausen, and L. Viennot. Investigating data broadcast performance in mobile adhoc networks. In *Proc. of WPMC*, Oct. 2002.
41. J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *Proc. of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.
42. H. Li and J. Wu. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of DialM*, 1999.
43. X. Lin and I. Stojmenovic. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks, 2001.
44. A. Lindgren, A. Doria, and O. Scheln. Probabilistic routing in intermittently connected networks. In *Proc. of ACM MobiHoc (poster session)*, Maryland, USA, June 2003.
45. X. Lu and W. Peng. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. of Mobihoc*, 2000.
46. X. Lu and W. Peng. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology*, 2002.
47. M. V. Marathe, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Geometry based heuristics for unit disk graphs, 1994.
48. A. G. Miklas, K. K. Gollu, S. Saroiu, K. P. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *Proc. of UBIComp*, Innsbruck, Austria, Sept 2007.
49. M. Mitzenmacher. Digital fountains: A survey and look forward. In *Proc. of IEEE Information Theory Workshop*, Texas, USA, Oct. 2004.
50. P. Nain, D. Towsley, B. Liu, and Z. Liu. Properties of random direction models. In *Proc. of IEEE Infocom*, Miami, FL, Mar. 2005.
51. M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Transactions on Information Theory*, 51(6), June 2005.
52. G. Neglia and X. Zhang. Optimal delay-power tradeoff in sparse delay tolerant networks: a preliminary study. In *Proc. of CHANTS*, Pisa, Italy, Sep. 2006.
53. S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proc. of MobiCom*, pages 151–162, New York, NY, USA, 1999. ACM.
54. J. Ott and D. Kutscher. Drive-thru internet: IEEE 802.11b for automobile users. In *Proc. of IEEE INFOCOM*, Hong Kong, Mar. 2004.
55. N. Sarafijanovic-Djukic, M. Piorowski, and M. Grossglauser. Island hopping: Efficient mobility-assisted forwarding in partitioned networks. In *IEEE SECON 2006*, Reston, VA, Sept. 2006.
56. V. Simon, L. Bacsardi, S. Szabo, and D. Miorandi. Bionets: A new vision of opportunistic networks. In *Proc. of IEEE WRECOM*, 2007.

57. T. Small and Z. J. Haas. The shared wireless infostation model: A new ad hoc networking paradigm. In *Proc. of ACM MobiHoc*, Annapolis, MD, USA, June 2003.
58. T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.
59. T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proc. of ACM Sigcomm Workshop on Delay Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.
60. R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.
61. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.
62. R. J. Vamsi, K. Paruchuria, and A. Durrësib. Optimized flooding protocol for ad hoc networks.
63. E. Varga, T. Csvorics, L. Bacsı̇rdi, M. Berces, V. Simon, and S. Szabi̇. Novel information dissemination solutions in biologically inspired networks. In *Proc. of ConTEL*, Zagreb Croatia, Jun. 2007.
64. L. Viennot, A. Qayyum, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical report, INRIA, 2000.
65. Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-coding based routing for opportunistic networks. In *Proc. of ACM Sigcomm Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.
66. J. Widmer and J.-Y. Le Boudec. Network coding for efficient communication in extreme networks. In *Proc. of ACM Sigcomm Workshop on Delay-Tolerant Networking*, Philadelphia, PA, USA, Aug. 2005.
67. B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proc. of MOBIHOC*, pages 194–205, 2002.
68. J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *Int. J. Found. Comput. Sci.*, 14(2):201–221, 2003.
69. X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *Proc. of Networking*, pages 827–839, Coimbra, Portugal, May 2006.
70. W. Zhao and M. Ammar. Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Proc. of the IEEE Workshop on Future Trends in Distributed Computing Systems*, Puerto Rico, May 2003.
71. W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of ACM Mobihoc*, Tokyo, Japan, May 2004.
72. W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005.

Part II

Paradigms from Physics

Free Deconvolution: from Theory to Practice

Florent Benaych-Georges¹ and Mérouane Debbah²

¹ LPMA/UMR 7599

Universit Paris 6

4, place Jussieu

75252 PARIS Cedex 05, France

florent.benaych@upmc.fr

² upélec

Ecole supérieure d'électricité

91192 - Gif sur Yvette Cedex, France

merouane.debbah@supelec.fr

Abstract. In this chapter, we review some important results on free deconvolution/random matrix theory and their application to wireless communications. In many situations, engineers are faced with the problem of extracting information from the network. As it will be shown, this corresponds in many respects to infer from the spectrum of functionals of random matrices with only a limited knowledge on the statistics of the matrix entries. In its full generality, the problem requires some sophisticated tools related to free probability and the explicit spectrum (complete information) can hardly be obtained (except for some trivial cases). Unfortunately, the advanced theoretical framework had led the community to the misconception that the tool has no practical application. This contribution takes the opposite view and shows how the free probability approach provides the right shift from theory to practice.

1 Introduction

Random matrix theory have been a part of advanced multivariate statistical analysis since the end of the 1920's. Random matrices were first proposed by Eugene Wigner in quantum mechanics where the statistics of experimentally measured energy levels of nuclei were explained in terms of the eigenvalues of random matrices. When Tsé [1] and Verdu [2] introduced nearly simultaneously random matrices in 1999 to model up-link unfaded CDMA systems equipped with certain type of receivers, random matrix theory entered the field of telecommunications. Until recently, in the field of information theory, simulations were widely believed to be the only means to optimise a given network. However simulations had to be very intensive and did not allow to single out parameters of interest easily. This changed when simultaneously in 1999, Tse [1] and Verdú [2] introduced tools of Random Matrix Theory in order to analyse multi-user systems, in the particular case of asymptotic performance of linear receivers for CDMA systems. They obtained explicit expressions for various measures of interest such as capacity or Signal to Interference plus Noise Ratio (SINR). Interestingly, it enables to single out the main parameters of interest that determine the performance in numerous models of communication systems with more or less involved models of attenuation [1,2,3,4,5]. In addition, these asymptotic results provide good approximations for the practical finite size case, as shown by simulations. A recent overview of Random Matrix Theory, centred on the applications to Wireless Communications, is given in the book by Tulino and Verdú [6].

Random matrices were first studied by statistical physicists. One of the first studies was done in 1928 by Wishart [7]. He computed the probability density of $\mathbf{v}_1\mathbf{v}_1^H + \dots + \mathbf{v}_n\mathbf{v}_n^H$ where \mathbf{v}_i are i.i.d. Gaussian vectors. His results are among the few available on finite dimensional matrices. The typical question is to characterise the distribution of (some of) the eigenvalues of random matrices. For finite matrix size this distribution itself is usually random. The real interest in random matrices surged when non-random limit distributions were derived for matrices whose dimensions tend to infinity, among others in 1955 by Wigner [8] and in 1967 by Marchenko and Pastur [9]. The introduction of the (Cauchy-)Stieltjes transform [10,11,12] then enabled to derive distributions for more involved expressions: correlation among the elements of the matrix, independent (non necessarily identically distributed) elements, and sums and products of random matrices. Random matrices are also particular non-commutative random variables.

Nowadays Random Matrix Theory is used in numerous domains, including but not limited to Riemann hypothesis, stochastic differential equations, condensed matter physics, statistical physics, chaotic systems, numerical linear algebra, neural networks, multivariate statistics, stock exchange analysis, and Information Theory.

Another question that naturally arises in cognitive random networks [13] is the following: "From a set of p noisy measurements, what can an intelligent device with n dimensions (time, frequency or space) extract in terms of useful information on the network? Moreover, once this information has been extracted, how can the terminal exploit (by capacity assessment, power allocation,...) that information?". It turns that these questions have recently found answers in the realm of free deconvolution. Cognitive Random Networks have been recently advocated as the next big evolution of wireless networks. The general framework is to design self-organising secure networks where terminals and base stations interact through cognitive sensing capabilities. The complexity of these systems requires some sophisticated tools based on free probability to make abstraction of the useless parameters. Free probability theory [14] is not a new tool but has grown into an entire field of research since the pioneering work of Voiculescu in the 1980's ([15,16,17,18]). However, the basic definitions of free probability are quite abstract and this has hinged a burden on its actual practical use. The original goal was to introduce an analogy to independence in classical probability that can be used for non-commutative random variables like matrices. These more general random variables are elements in what is called a *non-commutative probability space*, which we only introduce partially as our aim is to provide a more practical approach to these methods. Based on the moment/cumulant approach, the free probability framework has been quite successfully applied recently in the pioneering works [19,20] to extract information (where information in wireless networks is related to the eigenvalues of the random network) for very simple models i.e. the case where one of the considered matrices is unitarily invariant. This invariance has a special meaning in wireless networks and supposes that there is some kind of symmetry in the problem to be analysed. In this contribution, although focused on wireless communications, we show that the cumulant/moment approach can be extended to more general models and provide explicit algorithms to extract information (compute spectra) on the network. In this paper, we give an explicit relation between the spectra of random matrices $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$, $\mathbf{M}\mathbf{M}^*$ and $\mathbf{N}\mathbf{N}^*$, where \mathbf{M}, \mathbf{N} are large rectangular independent random matrices, at least one of them having a distribution which is invariant under multiplication, on any side, by any orthogonal matrix. This had already been done ([21,22,23]), but only in the case where \mathbf{M} or \mathbf{N} is Gaussian.

2 Definitions and Notations

We consider the general case of two independent real rectangular random matrices \mathbf{M}, \mathbf{N} , both having size $n \times p$. We shall suppose that n, p tend to infinity in such a way that n/p tends to a real number $\lambda \in (0, 1)$. We also suppose that at least one of these matrices has a distribution which is invariant by multiplication on any side by any orthogonal matrix. At last, we suppose that the *eigenvalue distributions* of $\mathbf{M}\mathbf{M}^*$ and $\mathbf{N}\mathbf{N}^*$ (i.e. the uniform distributions on their eigenvalues with multiplicity) both converge to non random probability measures. From a historical and purely mathematical perspective, people have focused on these types of random matrices because the invariance under actions of the orthogonal group is a – quite natural – notion of *isotropy*. The Gram³ assumption was mainly due to the fact that the eigenvalues (which are real and positive) are easier to characterise. From an engineering perspective, for a random network modelled by a matrix \mathbf{M} , the eigenvalues of $\mathbf{M}\mathbf{M}^*$ contain all the sufficient information to characterise the performance of the system. In fact, the eigenvalues relate mainly to the energy of the system. We shall explain how one can deduce, in a computational way, the limit eigenvalue distribution of $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$ from the limit eigenvalue distributions of $\mathbf{M}\mathbf{M}^*$ and $\mathbf{N}\mathbf{N}^*$. The underlying operation on probability measures is called the *rectangular free convolution with ratio* λ and denoted by \boxplus_λ in the literature ([24,25,26]). Our machinery will also allow the inverse operation, called *rectangular deconvolution with ratio* λ : the derivation of the eigenvalue distribution of $\mathbf{M}\mathbf{M}^*$ from the ones of $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$ and $\mathbf{N}\mathbf{N}^*$.

³ For a matrix \mathbf{M} , $\mathbf{M}\mathbf{M}^*$ is called the Gram matrix associated to \mathbf{M}

We shall also review some classical results of free probability and show how (as long as moments of the distributions are considered). One can, for \mathbf{A}, \mathbf{B} independent large square hermitian (or symmetric) random matrices (under some general hypothesis that will be specified):

- Derive the eigenvalue distribution of $\mathbf{A} + \mathbf{B}$ from the ones of \mathbf{A} and \mathbf{B} .
- Derive the eigenvalue distribution of \mathbf{AB} or of $\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}$ from the ones of \mathbf{A} and \mathbf{B} .

The previous operations are known in the literature as free convolutions ([27]), and denoted respectively by \boxplus, \boxtimes .

We will also see how one can:

- Deduce the eigenvalue distribution of \mathbf{A} from the ones of $\mathbf{A} + \mathbf{B}$ and \mathbf{B} .
- Deduce the eigenvalue distribution of \mathbf{A} from the ones of \mathbf{AB} or of $\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}$ and \mathbf{B} .

These last operations are called free deconvolutions ([21]) and denoted respectively by \boxminus, \boxdiv .

3 Information in random networks

In wireless intelligent random networks, devices are autonomous and should take optimal decisions based on their sensing capabilities. Of particular interest are information measures such as capacity, signal to noise ratio, estimation of powers or even topology identification. Information measures are usually related to the spectrum (eigenvalues) of the underlying network and not on the specific structure (eigenvectors). This entails many simplifications that make free deconvolution a very appealing framework for the design of these networks.

The fact that the spectrum of a stationary process is related to the information measure of the underlying process dates back to Kolmogorov [28]. One can show that the entropy rate of a stationary Gaussian stochastic process can be expressed as:

$$H = \log(\pi e) + \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(S(f)) df,$$

where S is the spectral density of the process. Hence, if one knows the auto-correlation of the process, one has therefore a full characterisation of the information contained in the process. Moreover, as a side result, one can also show that the entropy rate is also related to the minimum mean squared error of the best estimator of the process given the infinite past [29,30]. This remarkable result is of main interest for wireless communications as one can deduce one quantity from the other, especially as many receivers incorporate an MMSE (Minimum Mean Square Error) component. These results show the central importance of the auto-correlation function for Gaussian processes. In the discrete case when considering a random Gaussian vector \mathbf{x}_i of size n , the entropy rate per dimension (or differential entropy) is given by:

$$\begin{aligned} H &= \log(\pi e) + \frac{1}{n} \log \det(\mathbf{R}) \\ &= \log(\pi e) + \frac{1}{n} \sum_{i=1}^n \log(\lambda_i), \end{aligned} \quad (1)$$

where $\mathbf{R} = \mathbb{E}(\mathbf{x}_i \mathbf{x}_i^*)$ is the covariance matrix and λ_i the associated eigenvalues. The covariance matrix (and more precisely its eigenvalues) carries therefore all the information of Gaussian networks. The Gaussian property of these networks is due to the fact that the noise, the channel and the signalling is very often Gaussian. Hence, in order to get a reliable estimate of the rate (and in extension the capacity which is the difference between two differential entropies or any other measure which involves performance criteria), one needs to compute the eigenvalues of the covariance. For a number of observations p of the vector $\mathbf{x}_i, i = 1, \dots, p$, the covariance \mathbf{R} is usually estimated by:

$$\hat{\mathbf{R}} = \frac{1}{p} \sum_{i=1}^p \mathbf{x}_i \mathbf{x}_i^* \quad (2)$$

$$= \mathbf{R}^{\frac{1}{2}} \mathbf{S} \mathbf{S}^* \mathbf{R}^{\frac{1}{2}} \quad (3)$$

Here, $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_p]$ is an $n \times p$ i.i.d. zero mean Gaussian vector with entries of variance $\frac{1}{p}$. It turns out that in wireless random networks, the number of samples p is of the same order as n . This is mainly due to the fact that the network is highly mobile and the statistics are considered to be the same within a number p of samples, which restricts the use of classical asymptotic signal processing techniques. Therefore, information retrieval must be performed within a window of limited samples. The main advantage of free deconvolution techniques is that asymptotic results emerge at a much earlier stage than other techniques available up to now. The deconvolution framework comes here from the fact that we would like to invert equation (3) and express \mathbf{R} with respect to $\hat{\mathbf{R}}$. As we will show later on, this is in general not possible; however, one can compute the eigenvalues of \mathbf{R} knowing only the eigenvalues of $\hat{\mathbf{R}}$ (as the limiting eigenvalues of $\mathbf{S}\mathbf{S}^*$ are known to be the Marchenko-Pastur law). This is mainly due to the fact that due to our invariance assumption on one of the matrices (here $\mathbf{S}\mathbf{S}^*$), the eigenvector structure does not intervene in the final result. As a metaphorical viewpoint, the invariance assumption can be seen as freeing in some sense one matrix from the other by disconnecting their respective eigenspaces.

4 Eigenvalue distribution, joint eigenvalue distribution and moments

4.1 Eigenvalue distribution and joint eigenvalue distribution

It is important to note at this stage what we mean by eigenvalue distribution. In general, one would like to obtain the joint eigenvalue distribution of matrices or some marginal distribution with respect to the smallest eigenvalue for example. It turns out that in wireless communications, functionals of the eigenvalues are enough to characterise the measure of information or to retrieve information. Typically, as for the information rate, one is interested in

$$C_n = \frac{1}{n} \sum_{i=1}^n f(\lambda_i),$$

where f is a continuous function ($\log(1+x)$ for example)⁴. Note that C_n is a random variable (due to the fact that the eigenvalues λ_i are random) and can be expressed as:

$$C_n = \int f(\lambda) \frac{1}{n} \delta(\lambda - \lambda_i) d\lambda \quad (4)$$

$$= \int f(\lambda) d\rho_n(\lambda) \quad (5)$$

We call here ρ_n the eigenvalue distribution which turns out to be nothing else than a projection of the vector of eigenvalues $[\lambda_1, \dots, \lambda_n]$ into a single quantity. It is therefore less explicit than the joint eigenvalue distribution but is a sufficient statistics for our problem. ρ_n is also a random variable but as the dimension of the system grows, the behaviour of the eigenvalue distribution becomes deterministic in many cases. We will denote the asymptotic distribution ρ . Of course, other projections may also be of interest in other applications such as the maximum of the vector $[\lambda_1, \dots, \lambda_n]$.

4.2 Eigenvalue distribution and moments

Let us consider a probability measure ρ on the real line, which has moments of all order. We shall denote by $(m_k(\rho) := \int t^k d\rho(t))_{k \geq 0}$ the sequence of its moments. A given sequence of moments $\{m_k, k \geq 0\}$ may not uniquely determine the associated probability distribution. A trivial sufficient condition for this property to hold though is the existence of the moment generating function⁵. In any case, for computing the eigenvalue distribution when this condition is met, one needs to determine the moments of all orders.

⁴ In general, the function f should have other constraints (bounded) but for clarity reasons, we do not go into more details

⁵ A more sophisticated condition is the Carleman condition, which states that the sequence characterises a distribution if the following holds:

$$\sum_{i=1}^{\infty} m_{2i}^{-\frac{1}{2i}} = \infty$$

However, once again, practical applications show that the limiting eigenvalue distributions in wireless communications depends only a subset of parameters, typically:

$$d\rho(\lambda) = \frac{1}{L} \sum_{i=1}^L \delta(\lambda - \lambda_i),$$

where L is small and is related to the problem of interest (class of users with a given power in multi-user systems, number of scatterers in an environment, rank of the MIMO matrix in multiple antenna systems for example).

In this case, the moments are related to the eigenvalues by the following relations:

$$m_k(\rho) := \frac{1}{L} \sum_{i=1}^L \lambda_i^k. \quad (6)$$

As detailed in [31,19,32], one needs only to compute L moments to retrieve the eigenvalues in equation (6). This simplifies drastically the problems and favours a moment approach to the free deconvolution framework rather than deriving the explicit spectrum.

The *Newton-Girard Formula* [31] can be used to retrieve the eigenvalues from the moments. These formula state a relationship between the elementary symmetric polynomials

$$\Pi_j(\lambda_1, \dots, \lambda_L) = \sum_{i_1 < \dots < i_j \leq L} \lambda_{i_1} \cdots \lambda_{i_j}, \quad (7)$$

and

$$\begin{aligned} S_p(\lambda_1, \dots, \lambda_L) &= \sum_{i=1}^L \lambda_i^p \\ &= L \times m_p(\rho) \end{aligned}$$

through the recurrence relation

$$\begin{aligned} &(-1)^m m \Pi_m(\lambda_1, \dots, \lambda_L) \\ &+ \sum_{k=1}^m (-1)^{k+m} S_k(\lambda_1, \dots, \lambda_L) \Pi_{m-k}(\lambda_1, \dots, \lambda_L) = 0. \end{aligned} \quad (8)$$

Interestingly, the characteristic polynomial

$$(\lambda - \lambda_1) \cdots (\lambda - \lambda_L)$$

(which roots provides the eigenvalues of the associated matrix) can be fully characterized as its $L - k$ coefficient is given by: $(-1)^k \Pi_k(\lambda_1, \dots, \lambda_L)$. As $m_p(\rho)$ (we will show later on how these quantities can be computed) are known for $1 \leq p \leq L$, (8) can be used repeatedly to compute $\Pi_m(\lambda_1, \dots, \lambda_L)$, $1 \leq m \leq L$.

Note however that this method, exact to retrieve the eigenvalues from the successive moments of their distribution, might not be an optimal way to infer the unknown eigenvalues from the empirical moments of the underlying random matrices. This is discussed further subsequently.

4.3 Information plus Noise model

Example (3) is unfortunately rarely encountered in practice in wireless communications. The signal of interest \mathbf{s}_i is usually distorted by a medium, given by $\mathbf{m}_i = f(\mathbf{s}_i)$ where f is any function. Moreover, the received signal \mathbf{y}_i is altered by some additive noise vector \mathbf{n}_i (not necessarily Gaussian) but in many respect unitarily invariant (due to the fact that all the dimensions can be assumed to have the same importance). In this case, the model is known as the Information plus Noise model:

$$\mathbf{y}_i = \mathbf{m}_i + \mathbf{n}_i,$$

which can be rewritten in the following matrix form by stacking all the observations:

$$\mathbf{Y} = \mathbf{M} + \mathbf{N}. \quad (9)$$

The main challenge here is to infer on the eigenvalues of $\mathbf{M}\mathbf{M}^*$ when $\mathbf{Y}\mathbf{Y}^*$ and the eigenvalues of $\mathbf{N}\mathbf{N}^*$ are known (indeed, the noise can be measured by opening the receptor before the arrival of the signal \mathbf{m}). Once the spectrum of $\mathbf{M}\mathbf{M}^*$ has been computed (and depending on the function f), one can get the required information on the process s_i . Previous authors have addressed this problem through the Stieljes transform method [23] and other related methods for specific [33,34,35] models (typically, the case where \mathbf{N} has i.i.d. entries). In this work, we will show however that the cumulants/moments approach can be a very efficient tool and provide implementable algorithms for another class of models, i.e. those models based on unitarily invariant matrices.

5 Historical Perspective

Wigner [36] was interested in deriving the energy levels of nuclei. It turns out that energy levels are linked to the Hamiltonian operator by the following Schroedinger equation:

$$\mathbf{H}\phi_i = \mathbf{E}_i\phi_i$$

where

ϕ_i is the wave function

\mathbf{E}_i is the energy level

\mathbf{H} is the Hamiltonian

Hence, the energy levels of the operator \mathbf{H} is nothing else but the eigenvalues of the matrix representation of that operator. For a specific nuclei, finding the exact eigenvalues is a very intricate problem as the number of interacting particles increases. The genuine idea of Wigner was to replace the exact matrix by a large dimensional random matrix having the same properties. Hence, in some cases, the matrix can be replaced by the following Hermitian random matrix where the upper diagonal elements are i.i.d. generated with a binomial distribution.

$$\mathbf{H} = \frac{1}{\sqrt{n}} \begin{bmatrix} 0 & +1 & +1 & +1 & -1 & -1 \\ +1 & 0 & -1 & +1 & +1 & +1 \\ +1 & -1 & 0 & +1 & +1 & +1 \\ +1 & +1 & +1 & 0 & +1 & +1 \\ -1 & +1 & +1 & +1 & 0 & -1 \\ -1 & +1 & +1 & +1 & -1 & 0 \end{bmatrix}$$

It turns out as the dimension of the matrix increases, the eigenvalues of the matrix become more and more predictable irrespectively of the exact realisation of the matrix. This striking result enabled to determine the energy levels of many nuclei without considering the very specific nature of the interactions. In the following, we will provide the different steps of the proof which are of interest for understanding the moment approach.

6 Theoretical background

In the following, upper case and lower case boldface symbols will be used for matrices and column vectors, respectively. $(\cdot)^T$ will denote the transpose operator, $(\cdot)^*$ conjugation and $(\cdot)^H = ((\cdot)^T)^*$ Hermitian transpose. \mathbb{E} denotes the expectation operator.

Definition 1. Let $\mathbf{v} = [v_1, \dots, v_N]$ be a vector. Its empirical distribution is the function $F_N^{\mathbf{v}} : \mathbb{R} \rightarrow [0, 1]$ defined by:

$$F_N^{\mathbf{v}}(x) = \frac{1}{N} \#\{v_i \leq x \mid i = 1 \dots N\}.$$

corresponds to the cardinality operator. In other words, $F_N^{\mathbf{v}}(x)$ is the fraction of elements of \mathbf{v} that are inferior or equal to x . In particular, if \mathbf{v} is the vector of eigenvalues of a matrix \mathbf{V} , $F_N^{\mathbf{v}}$ is called the *empirical eigenvalue distribution* of \mathbf{V} .

A Wigner matrix is an $N \times N$ symmetric matrix \mathbf{H} with diagonal entries zero and upper-triangle entries i.i.d. zero mean and unit variance. As $N \rightarrow \infty$, the empirical eigenvalue distribution of $\frac{1}{\sqrt{N}}\mathbf{H}$ converges to the *semicircle law*:

$$f(\lambda) = \begin{cases} \frac{1}{2\pi}\sqrt{4-\lambda^2} & \text{if } |\lambda| \leq 2 \\ 0 & \text{if } |\lambda| \geq 2 \end{cases}$$

The semicircle law is plotted in Fig. 1, as well as the plot obtained by tracing the histogram of the eigenvalues of a *single* realisation of a 512×512 Wigner matrix, with i.i.d. Gaussian $\mathcal{N}(0, 1)$ distribution of the upper-triangle entries. The semicircle already provides a good approximation of the eigenvalue distribution in the finite size case. Note that even though the entries are theoretically not bounded, the distribution of the eigenvalues has a bounded support.

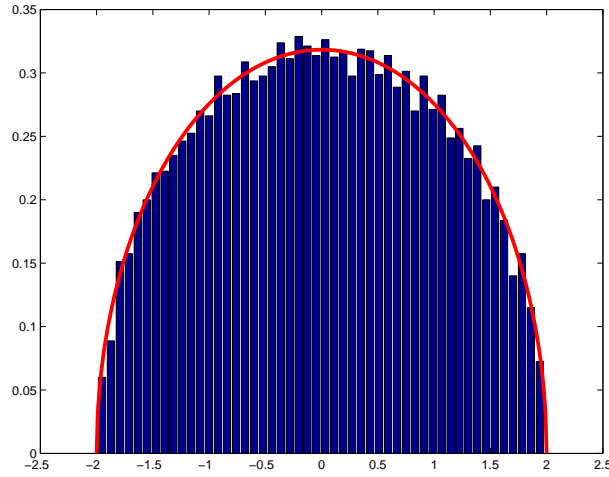


Fig. 1. Semicircle law and simulation for a 512×512 Wigner matrix.

Wigner matrices have quite a constrained symmetric form; it is in fact possible to obtain results for non-symmetric matrices. If \mathbf{H} is an $N \times N$ matrix with entries i.i.d. zero mean and variance 1, then the eigenvalues of $\frac{1}{\sqrt{N}}\mathbf{H}$ are uniformly distributed on the unit circle. This property is often referred to as Girko's full circle law.

The full circle law is plotted in Fig. 2, as well as the plot of the eigenvalues of a *single* realisation of a 512×512 random matrix, with i.i.d. Gaussian $\mathcal{N}(0, 1)$ distribution of the entries.

When non-square $N \times K$ matrices are under consideration, a common property of to ensure asymptotic convergence of the distribution is that the ratio of the dimension $\frac{K}{N}$ be kept constant. One of the first derivations of an explicit non-random limit distribution is due to Marchenko and Pastur . Let \mathbf{H} be an $N \times K$ matrix, with i.i.d. zero-mean complex entries with variance $\frac{1}{N}$ and fourth moments $O(\frac{1}{N^2})$. As $K, N \rightarrow \infty$, with $\frac{K}{N} \rightarrow \alpha$, the empirical eigenvalue distribution of $\mathbf{H}^H\mathbf{H}$ converges almost surely to a non-random limit distribution with density

$$f(x) = \left[1 - \frac{1}{\alpha}\right]^+ \delta(x) + \frac{\sqrt{[x-a]^+ [b-x]^+}}{2\pi\alpha x}$$

where $a = (1 - \sqrt{\alpha})^2$ and $b = (1 + \sqrt{\alpha})^2$.

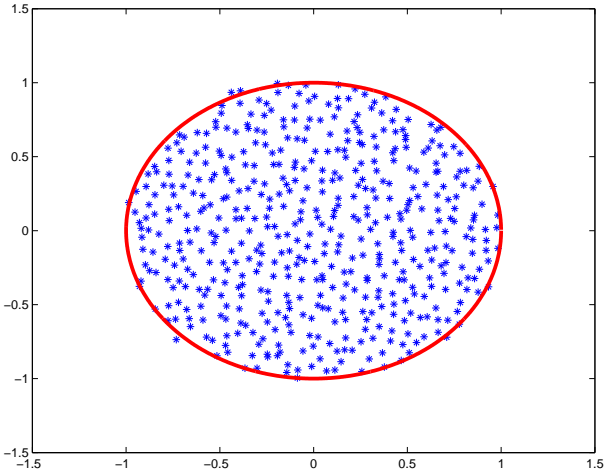


Fig. 2. Full circle law and simulation for a 512×512 matrix.

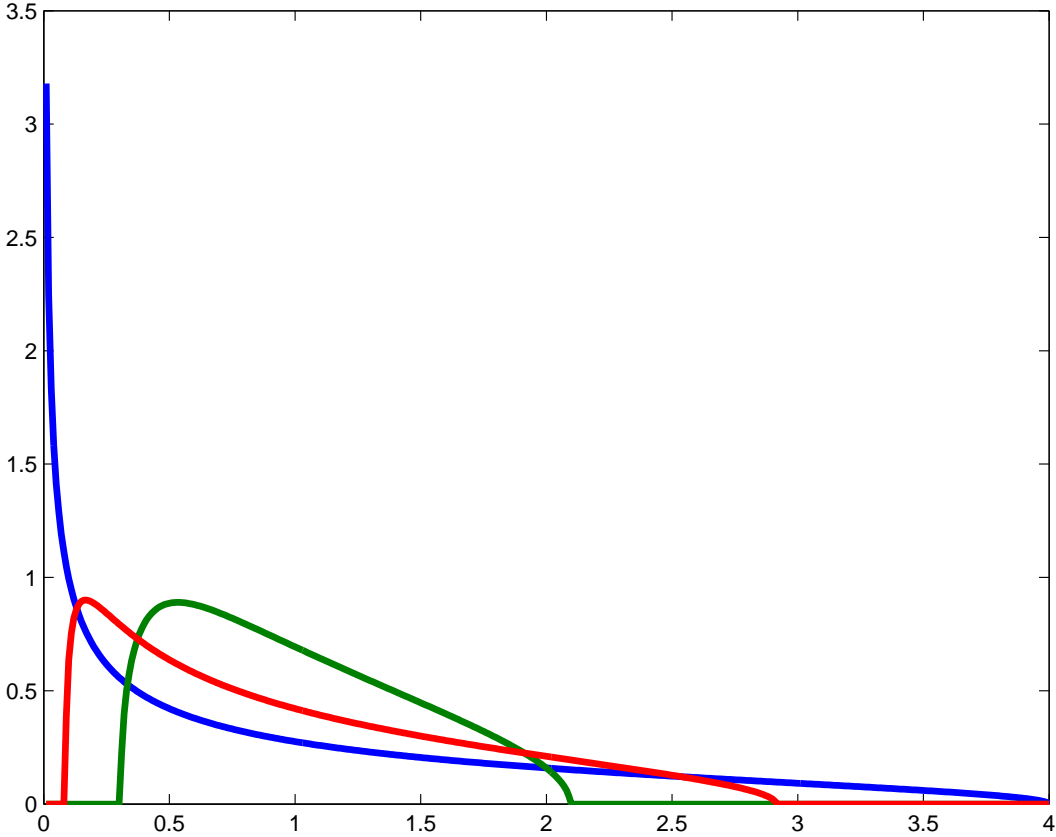


Fig. 3. Marchenko-Pastur density function for $\alpha = 1, 0.5, 0.2$.

The Marchenko-Pastur law is plotted in Fig. 3. The asymptotic analysis has an averaging effect: the limit distribution depends only on α , and not on the particular distribution of the entries of the matrices. The eigenvalues have a bounded support between $(1 - \sqrt{\alpha})^2$ and $(1 + \sqrt{\alpha})^2$.

In the following, we will review the basis tools needed to use random matrix theory results.

7 Moments approach

7.1 The semi-circular law

The main idea is to compute, as the dimension increases, the trace of the matrix \mathbf{H} at different powers. Typically, let

$$dF_N(\lambda) = \frac{1}{N} \sum_{i=1}^N \delta(\lambda - \lambda_i)$$

then the moments of the distribution are given by:

$$\begin{aligned} m_1^N &= \frac{1}{N} \text{trace}(\mathbf{H}) = \frac{1}{N} \sum_{i=1}^N \lambda_i = \int \lambda dF_N(\lambda) \\ m_2^N &= \frac{1}{N} \text{trace}(\mathbf{H})^2 = \int \lambda^2 dF_N(\lambda) \\ &\dots = \dots \\ m_k^N &= \frac{1}{N} \text{trace}(\mathbf{H})^k = \int \lambda^k dF_N(\lambda) \end{aligned}$$

Quite remarkably, as the dimension increases, the traces can be computed using combinatorial and non-crossing partitions techniques. All the moments converge to what is known as the Catalan numbers. In particular, we have:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \text{Trace}(\mathbf{H}^{2k}) &= \int_{-2}^2 x^{2k} f(x) dx \\ &= \frac{1}{k+1} C_k^{2k} \end{aligned}$$

Note that since the semi-circle law is symmetric, the odd moments vanish. More importantly, the only distribution which has all its moments equal to the Catalan number is known to be the semi-circular law given by:

$$f(x) = \frac{1}{2\pi} \sqrt{4 - x^2}$$

with $|x| \leq 2$. One can verify it directly by calculus based on recursion:

$$\begin{aligned} \alpha_{2k} &= \frac{1}{\pi} \int_{-2}^2 x^{2k} \sqrt{4 - x^2} dx \\ &= -\frac{1}{2\pi} \int_{-2}^2 \frac{-x}{\sqrt{4 - x^2}} x^{2k-1} (4 - x^2) dx \\ &= \frac{1}{2\pi} \int_{-2}^2 \sqrt{4 - x^2} (x^{2k-1} (4 - x^2))' dx \\ &= 4(2k - 1)\alpha_{2k-2} - (2k + 1)\alpha_{2k} \end{aligned}$$

In this way, the recursion is obtained:

$$\alpha_{2k} = \frac{2(2k - 1)}{k + 1} \alpha_{2k-2}$$

whose solution, together with $\alpha_2 = 1$ are the Catalan numbers. If one verifies that the i.i.d. entries of \mathbf{H} satisfy the Carleman condition, then the moments of the eigenvalue distribution of \mathbf{H} uniquely determine the distribution, and Wigner's semi-circle law is proven.

7.2 The Marchenko-Pastur Law

Let us give another example to understand the moments approach for a single random matrix. Suppose that one is interested in the empirical eigenvalue distribution of $\mathbf{H}\mathbf{H}^H$ where \mathbf{H} is $N \times K$ i.i.d. Gaussian variance $\frac{1}{N}$ with $\frac{K}{N} = \alpha$. In this case, in the same manner, the moments of this distribution are given by:

$$\begin{aligned}
 m_1^N &= \frac{1}{N} \text{trace}(\mathbf{H}\mathbf{H}^H) = \frac{1}{N} \sum_{i=1}^N \lambda_i \rightarrow 1 \\
 m_2^N &= \frac{1}{N} \text{trace}(\mathbf{H}\mathbf{H}^H)^2 = \frac{1}{N} \sum_{i=1}^N \lambda_i^2 \rightarrow 1 + \alpha \\
 m_3^N &= \frac{1}{N} \text{trace}(\mathbf{H}\mathbf{H}^H)^3 = \frac{1}{N} \sum_{i=1}^N \lambda_i^3 \rightarrow \alpha^2 + 3\alpha + 1
 \end{aligned}$$

It turns out that the only distribution which has the same moments is known to be the Marchenko-Pastur Law.

Remark: In many cases, one would obviously think that the eigenvalues of

$$\underbrace{\left[\begin{array}{c} \left[\begin{array}{c} \mathbf{H} \end{array} \right] \\ \left[\begin{array}{c} \mathbf{H}^H \end{array} \right] \end{array} \right]}_{K}$$

when $N \rightarrow \infty$, $K/N \rightarrow \alpha$ are equal to one. Indeed, asymptotically, all the diagonal elements are equal to one and the extra-diagonal elements are equal to zero. However, although the matrix "looks" like identity, it is not identity. Indeed, there are $N^2 - N$ extra-diagonal terms which tend to zero at a rate of $O(\frac{1}{N^2})$. Therefore, the distance of the matrix to the identity matrix (in the Froebenius norm sense) is not zero.

8 Freeness

The basic definitions of free probability are quite abstract, as the aim was to introduce an analogy to independence in classical probability that can be used for non-commutative random variables like matrices. These more general random variables are elements in what is called a *non-commutative probability space*. This can be defined by a pair (A, ϕ) , where A is a unital **-algebra* with unit I , and ϕ is a normalised (i.e. $\phi(I) = 1$) linear functional on A . The elements of A are called random variables. In all our examples, A will consist of $n \times n$ matrices or random matrices. For matrices, ϕ will be the normalised trace tr_n , defined by (for any $a \in A$)

$$tr_n(a) = \frac{1}{n} Tr(a) = \frac{1}{n} \sum_{i=1}^n a_{ii},$$

while for random matrices, ϕ will be the linear functional τ_n defined by

$$\tau_n(a) = \frac{1}{n} \sum_{i=1}^n E(a_{ii}) = E(tr_n(a)).$$

The unit in these **-algebras* is the $n \times n$ identity matrix \mathbf{I}_n . The non-commutative probability spaces considered will all be *tracial*, i.e. ϕ satisfies the trace property $\phi(ab) = \phi(ba)$. The analogy to independence is called freeness:

Definition 2. A family of unital **-sub-algebras* $(A_i)_{i \in I}$ will be called a *free family* if

$$\left\{ \begin{array}{l} a_j \in A_{i_j} \\ i_1 \neq i_2, i_2 \neq i_3, \dots, i_{n-1} \neq i_n \\ \phi(a_1) = \phi(a_2) = \dots = \phi(a_n) = 0 \end{array} \right\} \Rightarrow \phi(a_1 \cdots a_n) = 0. \tag{10}$$

A family of random variables a_i is called a *free family* if the algebras they generate form a free family.

One can note that the condition $i_1 \neq i_n$ is not included in the definition of freeness. This may seem strange since if ϕ is a trace and $i_1 = i_n$, we can rearrange the terms so that two consecutive terms in (10) come from the same algebra. If this rearranged term does not evaluate to zero through the definition of freeness, the definition of freeness would be inconsistent. It is not hard to show that this small issue does not cause an inconsistency problem. To see this, assume that (10) is satisfied for all indices where the circularity condition $i_1 \neq i_n$ is satisfied. We need to show that (10) also holds for indices where $i_1 = i_n$. By writing

$$a_n a_1 = (a_n a_1 - \phi(a_n a_1)I) + \phi(a_n a_1)I = b_1 + \phi(a_n a_1)I, \quad (11)$$

we can express $\phi(a_1 \cdots a_n) = \phi(a_n a_1 a_2 \cdots a_{n-1})$ as a sum of the two terms $\phi(b_1 a_2 \cdots a_{n-1})$ and $\phi(a_n a_1) \phi(a_2 \cdots a_{n-1})$. The first term is 0 by assumption, since $\phi(b_1) = 0$, $b_1 \in A_{i_n}$ and $i_n \neq i_{n-1}$. The second term $\phi(a_n a_1) \phi(a_2 \cdots a_{n-1})$ contributes with zero when $i_2 \neq i_{n-1}$ by assumption. If $i_2 = i_{n-1}$, we use the same splitting as in (11) again, but this time on $\phi(a_2 \cdots a_{n-1}) = \phi(a_{n-1} a_2 a_3 \cdots a_{n-2})$, to conclude that $\phi(a_2 \cdots a_{n-1})$ evaluates to zero unless $i_3 = i_{n-2}$. Continuing in this way, we eventually arrive at the term $\phi(a_{n/2} a_{n/2+1})$ if n is even, or the term $\phi(a_{(n+1)/2})$ if n is odd. The first of these is 0 since $i_{n/2} \neq i_{n/2+1}$, and the second is 0 by assumption.

Definition 3. We say that a sequence of random variables a_{n1}, a_{n2}, \dots in a probability spaces (A_n, ϕ_n) converges in distribution if, for any $m_1, \dots, m_r \in \mathbb{Z}$, $k_1, \dots, k_r \in \{1, 2, \dots\}$, we have that the limit $\phi_n(a_{nk_1}^{m_1} \cdots a_{nk_r}^{m_r})$ exists as $n \rightarrow \infty$. If these limits can be written as $\phi(a_{k_1}^{m_1} \cdots a_{k_r}^{m_r})$ for some non-commutative probability space (A, ϕ) and free random variables $a_1, a_2, \dots \in (A, \phi)$, we say that the a_{n1}, a_{n2}, \dots are asymptotically free.

Asymptotic freeness is a very useful concept for our purposes, since many types of random matrices exhibit asymptotic freeness when their sizes get large. For instance, consider random matrices $\frac{1}{\sqrt{n}}\mathbf{A}_{n1}, \frac{1}{\sqrt{n}}\mathbf{A}_{n2}, \dots$, where the \mathbf{A}_{ni} are $n \times n$ with all entries independent and standard Gaussian (i.e. mean 0 and variance 1). Then it is well-known [14] that the $\frac{1}{\sqrt{n}}\mathbf{A}_{ni}$ are asymptotically free. The limit distribution of the $\frac{1}{\sqrt{n}}\mathbf{A}_{ni}$ in this case is called the *circular law*, due to the asymptotic distribution of the eigenvalues of $\frac{1}{\sqrt{n}}\mathbf{A}_{ni}$: when $n \rightarrow \infty$, these get uniformly distributed inside the unit circle of the complex plane [37,38]. Equation (10) enables one to calculate the mixed moments of free random variables a_1 and a_2 . In particular, the moments of $a_1 + a_2$ and $a_1 a_2$ can be evaluated. In order to calculate $\phi((a_1 + a_2)^4)$, we multiply out $(a_1 + a_2)^4$, and use linearity and (10) to compute all $\phi(a_{i_1} a_{i_2} a_{i_3} a_{i_4})$ ($i_j = 1, 2$). For example, to calculate $\phi(a_1 a_2 a_1 a_2)$, we write it as

$$\begin{aligned} & \phi((a_1 - \phi(a_1)I) + \phi(a_1)I)((a_2 - \phi(a_2)I) + \phi(a_2)I) \\ & ((a_1 - \phi(a_1)I) + \phi(a_1)I)((a_2 - \phi(a_2)I) + \phi(a_2)I), \end{aligned}$$

and multiply it out as 16 terms. The term

$$\begin{aligned} & \phi((a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I) \\ & (a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)) \end{aligned}$$

is zero by (10). The term

$$\begin{aligned} & \phi((a_1 - \phi(a_1)I)\phi(a_2)I(a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)) \\ & = \phi(a_2)\phi((a_1 - \phi(a_1)I)(a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)) \end{aligned}$$

can be calculated by writing

$$b = (a_1 - \phi(a_1)I)(a_1 - \phi(a_1)I)$$

(which also is in the algebra generated by a_1), setting

$$b = (b - \phi(b)I) + \phi(b)I,$$

and using (10) again. The same procedure can be followed for any mixed moments.

When the sequences of moments uniquely identify probability measures (which is the case for compactly supported probability measures), the distributions of $a_1 + a_2$ and $a_1 a_2$ provide two probability measures, which depend only on the probability measures associated with the moments of a_1, a_2 . Therefore we can define two operations on the set of probability measures: *Additive free convolution*

$$\mu_1 \boxplus \mu_2 \tag{12}$$

for the sum of free random variables, and *multiplicative free convolution*

$$\mu_1 \boxtimes \mu_2 \tag{13}$$

for the product of free random variables. These operations can be used to predict the spectrum of sums or products of asymptotically free random matrices. For instance, if a_{1n} has an eigenvalue distribution which approaches μ_1 and a_{2n} has an eigenvalue distribution which approaches μ_2 , one has that the eigenvalue distribution of $a_{1n} + a_{2n}$ approaches $\mu_1 \boxplus \mu_2$, so that $\mu_1 \boxplus \mu_2$ can be used as an eigenvalue predictor for large matrices. Eigenvalue prediction for combinations of matrices is in general not possible, unless we have some assumption on the eigenvector structures. Such an assumption which makes random matrices fit into a free probability setting (and make therefore the random matrices free), is that of *uniformly distributed eigenvector structure* (i.e. the eigenvectors point in some sense in all directions with equal probability).

We will also find it useful to introduce the concepts of *additive and multiplicative free deconvolution*:

Definition 4. *Given probability measures μ and μ_2 . When there is a unique probability measure μ_1 such that*

$$\mu = \mu_1 \boxplus \mu_2, \mu = \mu_1 \boxtimes \mu_2 \text{ respectively,}$$

we will write

$$\mu_1 = \mu \boxminus \mu_2, \mu_1 = \mu \boxdiv \mu_2 \text{ respectively.}$$

We say that μ_1 is the additive free deconvolution (respectively multiplicative free deconvolution) of μ with μ_2 .

It is noted that the symbols presented here for additive and multiplicative free deconvolution have not been introduced in the literature previously. With additive free deconvolution, one can show that there always is a unique μ_1 such that $\mu = \mu_1 \boxplus \mu_2$. For multiplicative free deconvolution, a unique μ_1 exists as long as we assume non-vanishing first moments of the measures. This will always be the case for the measures we consider.

Some probability measures appear as limits for large random matrices in many situations. One important measure is the Marčenko Pastur law μ_c ([39] page 9), also known as the free Poisson distribution in free probability. It is known that μ_c describes asymptotic eigenvalue distributions of *Wishart* matrices. Wishart matrices have the form $\frac{1}{N} \mathbf{R} \mathbf{R}^H$, where \mathbf{R} is an $n \times N$ random matrix with independent standard Gaussian entries. μ_c appears as limits of such when $\frac{n}{N} \rightarrow c$ when $n \rightarrow \infty$, Note that the Marčenko Pastur law can also hold in the limit for non-Gaussian entries.

9 Sum of two random matrices

9.1 Scalar case: $X + Y$

Let us consider two independent random variables X, Y and suppose that we know the distribution of $X + Y$ and Y and would like to infer on the distribution of X . One way of doing that is to form the moment generating functions

$$M_X(s) = \mathbb{E}(e^{sX}), M_{X+Y}(s) = \mathbb{E}(e^{s(X+Y)}).$$

It is then immediate to see that

$$M_X(s) = M_{X+Y}(s) / M_Y(s).$$

The distribution of X can be recovered from $M_X(s)$. This task is however not always easy to perform as the inversion formula does provide an explicit expression. Note also that the distribution of $X + Y$ is the convolution of the distribution of X with the distribution of Y . Here again, the expression is not always straightforward to obtain. It is rather advantageous to express the independence in terms of moments of the distributions or even cumulants which we denote by C_k the cumulant of order k . The cumulants are defined, by the formula

$$C_k(X) := \frac{\partial^n}{\partial t^n} \Big|_{t=0} \log(\mathbb{E}(e^{tX})).$$

For independent random matrices, they behave additively with respect to the convolution, i.e. we have, for all $k \geq 0$,

$$C_k(X + Y) = C_k(X) + C_k(Y).$$

Recall that the moments of a random variable X are the numbers $m_n(X) = \mathbb{E}(X^n)$, $n \geq 1$. It happens that moments and cumulants of a random variable can easily be deduce from each other by the formula

$$\forall n \geq 1, m_n(X) = \sum_{p=1}^n \sum_{\substack{k_1 \geq 1, \dots, k_p \geq 1 \\ k_1 + \dots + k_p = n}} C_{k_1}(X) \cdots C_{k_p}(X).$$

Thus the derivation of the law of X from the ones of $X + Y$ and Y can be done by computing the cumulants of X by the formula $C_k(X) = C_k(X + Y) - C_k(Y)$ and then deducing the moments of X from its cumulants.

9.2 Additive free convolution \boxplus

Definition It has been proved by Voiculescu ([27]) that for $\mathbf{A}_n, \mathbf{B}_n$ independent large n by n hermitian (or symmetric) random matrices (both of them having iid entries, or one of them having a distribution which is invariant under conjugation by any orthogonal matrix), if the eigenvalue distributions of $\mathbf{A}_n, \mathbf{B}_n$ converge, as n tends to infinity, to some probability measures μ, ν , then the eigenvalue distribution of $\mathbf{A}_n + \mathbf{B}_n$ converges to a probability measure which depends only on μ, ν , which is called the *additive free convolution* of μ and ν , and which will be denoted by $\mu \boxplus \nu$.

Computation of $\mu \boxplus \nu$ by the moment/cumulants approach Let us consider a probability measure ρ on the real line, which has moments of all orders. We shall denote its moments by $m_n(\rho) := \int t^n d\rho(t)$, $n \geq 1$. (Note that in the case where ρ is the eigenvalue distribution of a $d \times d$ matrix \mathbf{A} , these moments can easily be computed by the formula: $m_n(\rho) = \frac{1}{d} \text{Tr}(\mathbf{A}^n)$, where Tr denotes the *trace*.) We shall associate to ρ another sequence of real numbers, $(\mathfrak{K}_n(\rho))_{n \geq 1}$, called its *free cumulants*. The sequences $(m_n(\rho))$ and $(\mathfrak{K}_n(\rho))$ can be deduced one from each other by the fact that the formal power series

$$K_\rho(z) := \sum_{n \geq 1} \mathfrak{K}_n(\rho) z^n \text{ and } M_\rho(z) := \sum_{n \geq 1} m_n(\rho) z^n \quad (14)$$

are linked by the relation

$$K_\rho(z(M_\rho(z) + 1)) = M_\rho(z). \quad (15)$$

Equivalently, for all $n \geq 1$, the sequences $(m_0(\rho), \dots, m_n(\rho))$ and $(\mathfrak{K}_1(\rho), \dots, \mathfrak{K}_n(\rho))$ can be deduced one from each other via the relations

$$m_0(\rho) = 1$$

$$m_n(\rho) = \mathfrak{K}_n(\rho) + \sum_{k=1}^{n-1} \mathfrak{K}_k(\rho) \sum_{\substack{l_1, \dots, l_k \geq 0 \\ l_1 + \dots + l_k = n-k}} m_{l_1}(\rho) \cdots m_{l_k}(\rho)$$

for all $n \geq 1$.

The additive free convolution can be computed easily with the free cumulants via the following characterisation (see [40]).

⌋ For μ, ν compactly supported, $\mu \boxplus \nu$ is the only law m such that for all $n \geq 1$,

$$\mathfrak{K}_n(m) = \mathfrak{K}_n(\mu) + \mathfrak{K}_n(\nu).$$

9.3 The additive free deconvolution \boxminus

The moments/cumulants method can also be useful to implement the free additive deconvolution. The *additive free deconvolution* of a measure m by a measure ν is (when it exists) the only measure μ such that $m = \mu \boxplus \nu$. In this case, μ is denoted by $m \boxminus \nu$. By Theorem 9.2, when it exists, $m \boxminus \nu$ is characterized by the fact that for all $n \geq 1$, $\mathfrak{K}_n(m \boxminus \nu) = \mathfrak{K}_n(m) - \mathfrak{K}_n(\nu)$.

10 Product of two random matrices

10.1 Scalar case: XY

Suppose now that we are given two classical random variables X, Y , assumed to be independent. How do we find the distribution of X when only the distributions of XY and Y are given? The solution is quite straightforward since $\mathbb{E}((XY)^k) = \mathbb{E}(X^k)\mathbb{E}(Y^k)$, so that $\mathbb{E}(X^k) = \mathbb{E}((XY)^k)/\mathbb{E}(Y^k)$. Hence, using the moments approach, one has a neat algorithm to compute all the moments of the distribution. The case of matrices is rather involved and is explained in the following.

10.2 The multiplicative free convolution \boxtimes

Definition It has been proved by Voiculescu ([27]) that for $\mathbf{A}_n, \mathbf{B}_n$ independent large n by n positive hermitian (or symmetric) random matrices (both of them having iid entries, or one of them having a distribution which is invariant under conjugation by any orthogonal matrix), if the eigenvalue distributions of $\mathbf{A}_n, \mathbf{B}_n$ converge, as n tends to infinity, to some probability measures μ, ν , then the eigenvalue distribution of $\mathbf{A}_n \mathbf{B}_n$, which is equal to the eigenvalue distribution of $\mathbf{A}^{\frac{1}{2}} \mathbf{B} \mathbf{A}^{\frac{1}{2}}$ converges to a probability measure which depends only on μ, ν , which is called the *multiplicative free convolution* of μ and ν , and which will be denoted by $\mu \boxtimes \nu$.

Computation of $\mu \boxtimes \nu$ by the moment/cumulants approach Let us consider a probability measure ρ on the $[0, +\infty)$, which is not the Dirac mass at zero and which has moments of all order. We shall denote by $(m_n(\rho) := \int t^n d\rho(t))_{n \geq 0}$ the sequence of its moments. We shall associate to ρ another sequence of real numbers, $(\mathfrak{s}_n(\rho))_{n \geq 0}$, which are the coefficients of what is called its *S-transform*. The sequences $(m_n(\rho))$ and $(\mathfrak{s}_n(\rho))$ can be deduced one from each other by the fact that the formal power series

$$S_\rho(z) := \sum_{n \geq 1} \mathfrak{s}_n(\rho) z^{n-1} \text{ and } M_\rho(z) := \sum_{n \geq 1} m_n(\rho) z^n \tag{16}$$

are linked by the relation

$$M_\rho(z) S_\rho(M_\rho(z)) = z(1 + M_\rho(z)). \tag{17}$$

Equivalently, for all $n \geq 1$, the sequences $(m_1(\rho), \dots, m_n(\rho))$ and $(\mathfrak{s}_1(\rho), \dots, \mathfrak{s}_n(\rho))$ can be deduced one from each other via the relations

$$\begin{cases} m_1(\rho) \mathfrak{s}_1(\rho) = 1, \\ m_n(\rho) = \sum_{k=1}^{n+1} \mathfrak{s}_k(\rho) \sum_{\substack{l_1, \dots, l_k \geq 1 \\ l_1 + \dots + l_k = n+1}} m_{l_1}(\rho) \cdots m_{l_k}(\rho). \end{cases} \tag{18}$$

rmq 101 Note that these equations allow computations which run faster than the ones already implemented (e.g. [22]), because those ones are based on the computation of the coefficients \mathfrak{s}_n via non-crossing partitions and the Kreweras complement, which use more machine time.

Example 102 As an example, it can easily be computed that for the Marčenko-Pastur law μ_λ , for all $n \geq 1$, $\mathfrak{s}_n(\mu_\lambda) = (-\lambda)^{n-1}$.

The multiplicative free convolution can be computed easily with the free cumulants via the following characterisation ([40]).

‡ For μ, ν compactly supported probability measures on $[0, \infty)$, non of them being the Dirac mass at zero, $\mu \boxtimes \nu$ is the only law m such that $S_m = S_\mu S_\nu$, i.e. such that for all $n \geq 1$,

$$\mathfrak{s}_n(m) = \sum_{\substack{k,l \geq 1 \\ k+l=n+1}} \mathfrak{s}_k(\mu) \mathfrak{s}_l(\nu).$$

The algorithm for the computation of the spectrum of the product of two random matrices following from this theorem has been implemented. It is presented in the following paragraph 10.3.

10.3 The multiplicative free deconvolution

The moments/cumulants method can also be useful to implement the multiplicative free deconvolution. The *multiplicative free deconvolution* of a measure m by a measure ν is (when it exists) the only measure μ such that $m = \mu \boxtimes \nu$. In this case, μ is denoted by $m \boxminus \nu$. By theorem 10.2, when it exists, $m \boxminus \nu$ is characterized by the fact that for all $n \geq 1$,

$$\mathfrak{s}_n(m \boxminus \nu) \mathfrak{s}_1(\nu) = \mathfrak{s}_n(m) - \sum_{k=1}^{n-1} \mathfrak{s}_k(m \boxminus \nu) \mathfrak{s}_{n+1-k}(\nu).$$

Hence this operation, very useful to denoise a signal, can be easily implemented.

11 $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$

11.1 Main result

In this section, we still consider two independent rectangular random matrices \mathbf{M}, \mathbf{N} , both having size $n \times p$. We shall suppose that n, p tend to infinity in such a way that n/p tends to a real number $\lambda \in [0, 1]$. We also suppose that at least one of these matrices has a distribution which is invariant by multiplication on both sides by any orthogonal (or unitary, in the case where the matrices are not real but complex) matrix. At last, we suppose that the *eigenvalue distributions* of $\mathbf{M}\mathbf{M}^*$ and $\mathbf{N}\mathbf{N}^*$ (i.e. the uniform distributions on there eigenvalues with multiplicity) both converge to non random probability measures. Here, we shall denote by respectively σ, τ the limit eigenvalue distributions of $\mathbf{M}\mathbf{M}^*$ and $\mathbf{N}\mathbf{N}^*$.

Note that in the previously presented results, the case of the limit eigenvalue distribution of $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$ has not been treated. The reason is that these results rely on the works of Voiculescu, who “only” found out a general way to compute the limit normalised trace of product of independent *square* random matrices with large dimension, which is all we need to compute the moments of the eigenvalue distribution of either $\mathbf{M}\mathbf{M}^* + \mathbf{N}\mathbf{N}^*$ or $\mathbf{M}\mathbf{M}^*\mathbf{N}\mathbf{N}^*$ (because in these expression, both \mathbf{M} and \mathbf{N} are always followed by their adjoints), but which is not enough to compute the moments of the eigenvalue distribution of $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$. In a recent work ([24]), we generalised Voiculescu’s work to rectangular random matrices, which allowed him to prove that, under the hypothesis made here, the eigenvalue distribution of $(\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*$ converges to a probability measure which only depends on σ, τ and λ , and will be denoted by $\sigma \boxplus_\lambda^+ \tau$.

rmq 111 The symmetric square root⁶ of the distribution $\sigma \boxplus_\lambda^+ \tau$ is be called the *rectangular free convolution with ratio λ of the symmetric square roots* $\sqrt{\sigma}, \sqrt{\tau}$ of σ and τ , and denoted by $\sqrt{\sigma} \boxplus_\lambda \sqrt{\tau}$. The operation \boxplus_λ is, rather than \boxplus_λ^+ , the one we introduced in [24]. It is essentially equivalent to \boxplus_λ , as we explain in the footnote below.

⁶ For any probability measure ρ on $[0, \infty)$, the *symmetric square root* of ρ , denoted by $\sqrt{\rho}$, is the only symmetric probability measure on the real line which push-forward by the $t \mapsto t^2$ function is ρ . Note that ρ is completely determined by $\sqrt{\rho}$, and *vice versa*. In the theoretical paper [24], the use of symmetric measures was more appropriate, that’s why we chose to work with symmetric square roots of measures. However, in the present paper, where we try to present how *practically*, matrices are working, we shall not symmetrised distributions.

As for \boxplus , we shall see two ways to compute \boxplus_λ : the first one is accomplished *via* the moments and is easy to implement, and the second one relays on analytic functions and is practically working in very few cases.

11.2 Computing \boxplus_λ

Let us fix $\lambda \in [0, 1]$ and consider a probability measure ρ on $[0, +\infty)$, which has moments of all orders. We shall denote by $(m_n(\rho) := \int t^n d\rho(t))_{n \geq 0}$ the sequence of its moments. We shall associate to ρ another sequence of real numbers, $(c_n(\rho))_{n \geq 1}$, depending on λ , called its *rectangular free cumulants*⁷ with ratio λ , defined by the fact that the sequences $(m_n(\rho))$ and $(c_n(\rho))$ can be deduced one from each other by the relation

$$C_\rho[z(\lambda M_{\rho^2}(z) + 1)(M_{\rho^2}(z) + 1)] = M_{\rho^2}(z) \tag{19}$$

between the power series

$$C_\rho(z) := \sum_{n \geq 1} c_n(\rho) z^n \text{ and } M_{\rho^2}(z) := \sum_{n \geq 1} m_n(\rho) z^n. \tag{20}$$

Equivalently, for all $n \geq 1$, the sequences $(m_0(\rho), \dots, m_n(\rho))$ and $(c_1(\rho), \dots, c_n(\rho))$ can be deduced from one another via the relations (involving an auxiliary sequence $(m'_0(\rho), \dots, m'_n(\rho))$)

$$\begin{aligned} m_0(\rho) &= m'_0(\rho) = 1, \\ \forall n \geq 1, \quad m'_n(\rho) &= \lambda m_n(\rho), \\ \forall n \geq 1, \quad m_n(\rho) &= \\ c_n(\rho) + \sum_{k=1}^{n-1} c_k(\rho) &\sum_{\substack{l_1, l'_1, \dots, l_k, l'_k \geq 0 \\ l_1 + l'_1 + \dots + l_k + l'_k = n-k}} m_{l_1}(\rho) m'_{l'_1}(\rho) \cdots m_{l_k}(\rho) m'_{l'_k}(\rho). \end{aligned}$$

Example 112 As an example, it is proved in [41] that the law μ_λ has rectangular free cumulants with ratio λ given by $c_n(\mu_\lambda) = \delta_{n,1}$ for all $n \geq 1$.

The additive free convolution can be computed easily with the free cumulants via the following characterisation ([24]).

Theorem 113 For σ, τ compactly supported, $\sigma \boxplus_\lambda^+ \tau$ is the only distribution m such that for all $n \geq 1$, $c_n(m) = c_n(\sigma) + c_n(\tau)$.

11.3 The rectangular free deconvolution

The moments/cumulants method can also be useful to implement the rectangular free deconvolution. The *rectangular free deconvolution with ratio λ* of a probability measure m on $[0, +\infty)$ by a measure τ is (when it exists) the only measure σ such that $m = \sigma \boxplus_\lambda^+ \tau$. In this case, σ is denoted by $m \boxminus_\lambda \tau$. By Theorem 113, when it exists, $m \boxminus_\lambda \tau$ is characterized by the fact that for all $n \geq 1$,

$$c_n(m \boxminus_\lambda \tau) = c_n(m) - c_n(\tau).$$

Hence this operation can also be very easily implemented in software.

⁷ Note that again, in [24], these numbers were not called rectangular free cumulants of ρ , but of its symmetrized square root.

11.4 Discussion

Case where $\lambda = 0$ It is proved in [24] that if $\lambda = 0$, then $\boxplus_\lambda^+ = \boxplus$.

Concretely, it means that if \mathbf{M}, \mathbf{N} are independent $n \times p$ random matrices which dimensions n, p both tend to infinity such that $n/p \rightarrow 0$, then (under the hypothesis that \mathbf{M} or \mathbf{N} is invariant, in distribution, under multiplication by unitary matrices)

$$\begin{aligned} & \text{eigenvalue distribution}((\mathbf{M} + \mathbf{N})(\mathbf{M} + \mathbf{N})^*) \\ & \simeq \text{eigenvalue distribution}(\mathbf{M}\mathbf{M}^* + \mathbf{N}\mathbf{N}^*). \end{aligned}$$

Case where $\lambda = 1$ It is proved in [24] that if $\lambda = 1$, then for all σ, τ probability measures on $[0, +\infty)$, $\sigma \boxplus_\lambda^+ \tau$ is the push forward by the function $t \mapsto t^2$ of the free convolution $\sqrt{\sigma} \boxplus \sqrt{\tau}$ of the symmetrised square roots $\sqrt{\sigma}, \sqrt{\tau}$ of σ and τ .

An analytic tool: the rectangular R -transform $\lambda \in [0, 1]$ is still fixed. For τ probability measure on $[0, +\infty)$, we shall define an analytic function $C_\tau(z)$ in a neighbourhood of zero (to be more precise, in a neighbourhood of zero in $\mathbb{C} \setminus \mathbb{R}^+$) which, in the case where τ is compactly supported, has the following series expansion:

$$C_\tau(z) = \sum_{n \geq 1} c_n(\tau) z^n. \quad (21)$$

It implies, by Theorem 113, that for all compactly supported probability measures σ, τ ,

$$C_{\sigma \boxplus_\lambda^+ \tau}(z) = C_\sigma(z) + C_\tau(z). \quad (22)$$

Hence the analytic transform $\rho \mapsto C_\rho$ somehow "linearises" the binary operation \boxplus_λ^+ on the set of probability measures on $[0, +\infty)$. The analytic function $\tau \mapsto C_\tau$ is called the *rectangular R -transform*⁸ with ratio λ of ρ .

It happens, as we present it bellow, that for any probability measure ρ on $[0, +\infty)$, C_ρ can be computed in a direct way, without using the definition of (21), by the resolution of an equation.

Let us define $M_\rho(z) = \int_{t \in \mathbb{R}^+} \frac{zt}{1-zt} d\rho(t)$. Then the analytic function C_ρ is defined in a neighbourhood of zero (in $\mathbb{C} \setminus \mathbb{R}^+$) to be the solution of

$$C_\rho[z(\lambda M_\rho(z) + 1)(M_\rho(z) + 1)] = M_\rho(z). \quad (23)$$

which tends to zero at zero.

To give a more explicit definition of C_ρ , let us define

$$H_\rho(z) = z(\lambda M_\rho(z) + 1)(M_\rho(z) + 1).$$

Then

$$C_\rho(z) = U \left(\frac{z}{H_\rho^{-1}(z)} - 1 \right)$$

with

$$U(z) = \begin{cases} \frac{-\lambda - 1 + ((\lambda + 1)^2 + 4\lambda z)^{\frac{1}{2}}}{2\lambda} & \text{if } \lambda \neq 0, \\ z & \text{if } \lambda = 0, \end{cases}$$

where $z \mapsto z^{\frac{1}{2}}$ is the analytic version of the square root defined on $\mathbb{C} \setminus \mathbb{R}^-$ such that $1^{\frac{1}{2}} = 1$ and H_ρ^{-1} is the inverse (in the sense of the composition) of the function H_ρ .

⁸ Again, to make the notations of this paragraph coherent with the ones of the paper [24], where the rectangular machinery was build, one needs to use the duality between measures on $[0, +\infty)$ and their symmetrized square roots, which are symmetric measures on the real line.

To recover ρ from C_ρ , one has to go the inverse way:

$$H_\rho^{-1}(z) = \frac{z}{(\lambda C_\rho(z) + 1)(C_\rho(z) + 1)}$$

and

$$M_\rho(z) = U \left(\frac{H_\rho(z)}{z} - 1 \right),$$

from which one can easily recover ρ , via its Stieltjes transform.

Note that all this method is working for non-compactly supported probability measures, and that (22) is valid for any pair of symmetric probability measures.

As for \boxplus and \boxtimes , analytic functions give us a new way to compute the multiplicative free convolution of two symmetric probability measures τ, σ . However, as for \boxplus and \boxtimes , the operations which are necessary in this method (the inversion of certain functions, the extension of certain analytic functions) are almost always impossible to realise practically. However, in the following example, computations are possible.

Example 114 Suppose $\lambda > 0$. Then $\delta_1 \boxplus_\lambda^+ \delta_1$ has support $[(2 - \kappa), (2 + \kappa)]$ with $\kappa = 2(\lambda(2 - \lambda))^{1/2} \in (0, 2)$, and it admits a density with formula

$$\frac{[\kappa^2 - (x - 2)^2]^{1/2}}{\pi \lambda x(4 - x)} \tag{24}$$

on its support.

This means that if \mathbf{A} is an $n \times p$ matrix with ones on the diagonal and zeros everywhere else, and \mathbf{U}, \mathbf{V} are random $n \times n, p \times p$ orthogonal matrices with Haar distribution, then as n, p tend to infinity such that $n/p \rightarrow \lambda$,

eigenvalue distribution $(\mathbf{A} + \mathbf{UAV})(\mathbf{A} + \mathbf{UAV})^*$ has density

$$\simeq \frac{[\kappa^2 - (x - 2)^2]^{1/2}}{\pi \lambda x(4 - x)} \text{ on } [2 - \kappa, 2 + \kappa].$$

Indeed, $\frac{[\kappa^2 - (x - 2)^2]^{1/2}}{\pi \lambda x(4 - x)}$ is the density of the square of a random variable with density (24).

12 Examples in Wireless Random Networks

12.1 Topology information

The simplest example of use of the deconvolution framework is the case where the functional $f(\lambda)$ of the population eigenvalues under study is the identity. This case is of practical interest when one performs channel sounding measurements. The transmitter sends an impulse on a given band to sound the environment. The channel response (or more precisely its power delay profile through the covariance of the received signal) contains information on the structure of the environment. By appropriate ray tracing techniques, localisation can be performed with a single receiver. The time-delayed channel impulse response can be written as:

$$x(\tau) = \sum_{k=1}^L \sigma_k s_k g(\tau - \tau_k),$$

where s_k are zero mean unit Gaussian variables and σ_k are their associated variances (due to the topology), L represent the total number of scatterers and g is the transmit filter. In the frequency domain, the received vector for a given frequency f_i in the presence of noise, can be written:

$$y_i = x_i + n_i,$$

where $x_i = \sum_{k=1}^L s_k G(f_i) e^{-j2\pi f_i \tau_k}$. In matrix form,

$$\mathbf{y} = \mathbf{R}^{\frac{1}{2}} \mathbf{s} + \mathbf{n},$$

where $\mathbf{R}^{\frac{1}{2}} = \mathbf{G}\boldsymbol{\theta}\boldsymbol{\Sigma}$. Here, \mathbf{G} is a diagonal matrix with entries $G(f_i)$, $\boldsymbol{\theta}$ is $n \times L$ matrix with entries $e^{-j2\pi f_i \tau_k}$ and $\boldsymbol{\Sigma}$ is a diagonal matrix with entries σ_k . \mathbf{s} and \mathbf{n} are respectively $L \times 1$ and $n \times 1$ zero mean unit variance Gaussian vectors. The free deconvolution framework enables to infer on the L non-zero eigenvalues of \mathbf{R} and therefore σ_k as suggested in [19].

12.2 Capacity and SINR estimation

In the case of cognitive TDD (Time Division Duplex) MIMO systems (the transmitter and the receiver have multi-antenna elements), the receiver would like to infer on the rate based only on the knowledge of the variance of the noise σ^2 , but without any training systems and using only p samples. The TDD mode here enables channel reciprocity by providing the same rate on both ends. The received signal can be written as:

$$\mathbf{y}_i = \mathbf{H}\mathbf{s}_i + \mathbf{n}_i,$$

where \mathbf{H} is the $n \times n$ MIMO matrix. The information rate is given by [42]:

$$\begin{aligned} C &= \log \det \left(\mathbf{I} + \frac{1}{\sigma^2} \mathbf{H}\mathbf{H}^* \right) \\ &= \sum_{i=1}^n \log(1 + \lambda_i). \end{aligned} \quad (25)$$

One can also be interested in the estimation of the SINR (Signal to Interference plus Noise Ratio) at the output of the MMSE receiver (if Bit Error Rate requirements are imposed) which is asymptotically given by [43]:

$$\begin{aligned} \text{SINR} &= \frac{1}{n} \text{trace} \left(\mathbf{H}\mathbf{H}^* + \sigma^2 \mathbf{I} \right)^{-1} \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{\lambda_i + \sigma^2} \end{aligned}$$

In both cases, the number of non-zero eigenvalues is also limited to L in general as the medium (matrix \mathbf{H}) provides only a finite number of degrees of freedom. One can compute these eigenvalues by using the free deconvolution framework on $\mathbf{Y}\mathbf{Y}^*$.

12.3 Power estimation

In TDD heterogeneous systems where a terminal is connected to several base stations, determining the power of the signal received from each base station is important as it will induce the adequate rate splitting between the different base stations. Suppose that each base station in the down-link has a given signature vector of size $n \times 1$ \mathbf{h}_k (OFDM, CDMA) with random i.i.d. components, the received signal can be written as:

$$\mathbf{y} = \sum_{k=1}^L \mathbf{h}_k \sqrt{P_k} s_k + \mathbf{n}$$

where P_k is the power received from each base station. L is the number of base stations, s_k is the signal transmitted by base station k and \mathbf{y} and \mathbf{n} are respectively the $n \times 1$ received signal and additive noise. It turns out here once again that one can infer on the powers P_k knowing only $\mathbf{Y}\mathbf{Y}^*$ as shown in [32].

13 The Stieltjes Transform approach

Let μ be a probability measure on \mathbb{R} . Its Stieltjes transform

$$G_\mu(z) = \int_{-\infty}^{+\infty} \frac{d\mu(t)}{t-z}$$

is defined on \mathbb{C}/\mathbb{R} . When z lies in the upper half-plane $\mathbb{C}^+ = (z \in \mathbb{C} : \text{Im}(z) > 0)$, the transform $G_\mu(z)$ is an analytic function in \mathbb{C}^+ possessing the following properties:

$$G_\mu(\mathbb{C}^+) \subset \mathbb{C}^+ \text{ and } |G_\mu(z)| \leq \frac{1}{\text{Im}(z)}$$

Let μ be compactly supported. Then $G_\mu(z)$ is analytic in a neighbourhood of ∞ . Since $(z-t)^{-1} = \sum_{k=0}^{\infty} t^k z^{-k-1}$, it is obvious that $G_\mu(z)$ has the following expansion at $z = \infty$:

$$-G_\mu(z) = z^{-1} + \sum_{k=0}^{\infty} m_k(\mu) z^{-k-1}.$$

where $m_k(\mu) = \int t^k d\mu(t) (k \in \mathbb{Z}^+)$.

Writing $z = x + iy$, it is possible to recover μ from its Cauchy transform up to a factor. When μ is absolutely continuous with respect to the Lebesgue measure, its density $f(x)$ is given by:

$$f(x) = +\frac{1}{\pi} \lim_{y \rightarrow 0} \text{Im} G_\mu(x + iy)$$

The random matrix theory in consideration here (see [44,11] for more details) deals with the limiting distribution of random Hermitian matrices of the form $A + WDW^H$. Here, $W(N \times K), D(K \times K)$, and $A(N \times N)$ are independent, with W containing i.i.d. entries having finite second order moments, T is diagonal with real entries, A is Hermitian and $K/N \rightarrow \alpha > 0$ as $N \rightarrow \infty$. The behaviour is expressed in terms of the limiting distribution function F^{A+WDW^H} of the eigenvalues of $A + WDW^H$ (i.e. $F^{A+WDW^H}(x)$ is the proportion of eigenvalues of $A + WDW^H \leq x$). The remarkable result of random matrix theory is the convergence in some sense, of $F^{A+WDW^H}(x)$ to a non random F .

The papers vary in the assumption on T, W and A . We will only take here the case of interest.

Theorem 1. *Let A be a $N \times N$ hermitian matrix, nonrandom, for which F^A converges weakly as $N \rightarrow \infty$ to a distribution function \mathbb{A} . Let F^D converges weakly to a nonrandom probability distribution function denoted \mathbb{D} . Suppose the entries of $\sqrt{N}W$ i.i.d. for fixed N with unit variance (sum of the variances of the real and imaginary parts in the complex case). Then the eigenvalue distribution of $A + WDW^H$ converges weakly to a deterministic F . Its Stieltjes transform $G(z)$ satisfies the equation:*

$$G(z) = G_{\mathbb{A}} \left(z - \alpha \int \frac{\tau d\mathbb{T}(\tau)}{1 + \tau G(z)} \right) \tag{26}$$

13.1 A theoretical application example

We give hereafter the eigenvalue distribution of matrices defined by:

$$\mathbf{W}_{N,K} \mathbf{W}_{N,K}^H + \sigma^2 \mathbf{I}$$

where $\sqrt{N} \mathbf{W}_{N,K}$ is a $N \times K$ matrix with i.i.d. entries with zero mean and variance one and $N \rightarrow \infty$ such as $\frac{K}{N} \rightarrow \alpha$ fixed. This example is intended to show to what extent (in terms of the matrix dimension) theoretical and practical results fit. Moreover, we give the basic machinery that the reader can use to derive any limiting eigenvalue distribution of matrices defined as in Theorem 1.

Denote $A = \sigma^2 \mathbf{I}$ and $D = \mathbf{I}_{K,K}$. In this case, $d\mathbb{A}(x) = \delta(x - \sigma^2)$ and $d\mathbb{D}(x) = \delta(x - 1)$. Applying theorem 1, the following result is obtained:

$$G(z) = G_{\sigma^2 \mathbf{I}} \left(z - \alpha \int \frac{\tau \delta(\tau - 1) d\tau}{1 + \tau G(z)} \right) \quad (27)$$

$$= G_{\sigma^2 \mathbf{I}} \left(z - \frac{\alpha}{1 + G(z)} \right) \quad (28)$$

$$= \int \frac{\delta(\sigma^2 - \lambda) d\lambda}{\lambda - z + \frac{\alpha}{1 + G(z)}} \quad (29)$$

$$= \frac{1}{\sigma^2 - z + \frac{\alpha}{1 + G(z)}}. \quad (30)$$

$G_{\sigma^2 \mathbf{I}}(z)$ is the Cauchy transform of the eigenvalue distribution of matrix $\sigma^2 \mathbf{I}$. The solution of the second order Equation 30 yields:

$$G(z) = \frac{1 - \alpha}{2(\sigma^2 - z)} - \frac{1}{2} + \frac{1}{2(\sigma^2 - z)} \sqrt{((\sigma^2 - z + \alpha - 1)^2 + 4(\sigma^2 - z))}.$$

The asymptotic eigenvalue distribution is therefore given by:

$$f(\lambda) = \begin{cases} [1 - \alpha]^+ \delta(x) + \frac{\alpha}{\pi(\lambda - \sigma^2)} \sqrt{\lambda - \sigma^2 - \frac{1}{4}(\lambda - \sigma^2 + 1 - \alpha)^2} & \text{if } \sigma^2 + (\sqrt{\alpha} - 1)^2 \leq \lambda \leq \sigma^2 + (\sqrt{\alpha} + 1)^2 \\ 0 & \text{otherwise} \end{cases}$$

Where $\delta(x)$ is a unit point mass at 0 and $[z]^+ = \max(0, z)$.

A remarkable result is that the eigenvalues of such matrices have a compact support while the entries may take any value (subject to a zero mean and variance one distribution). We plotted the theoretical and practical eigenvalue distribution for various size matrices. Only one realisation of matrix $\mathbf{W}_{N,K} \mathbf{W}_{N,K}^H + \sigma^2 \mathbf{I}$ is studied. As the dimensions increase, the practical results fit extremely well the theoretical results as shown in fig.A.1 and fig.A.2.

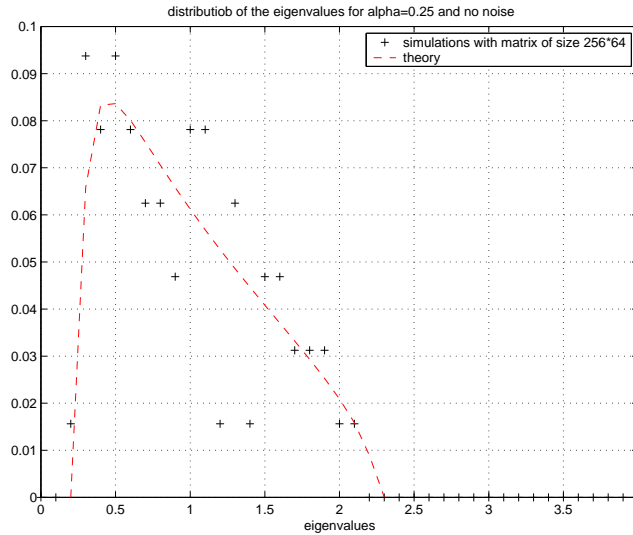


Fig. 4. matrix size: 256*64, no noise, alpha=0.25.

13.2 A wireless communication example

Most of the information theoretic literature focuses on vector memory-less channels of the form:

$$\mathbf{Y} = \mathbf{H}\mathbf{s} + \mathbf{n}. \quad (31)$$

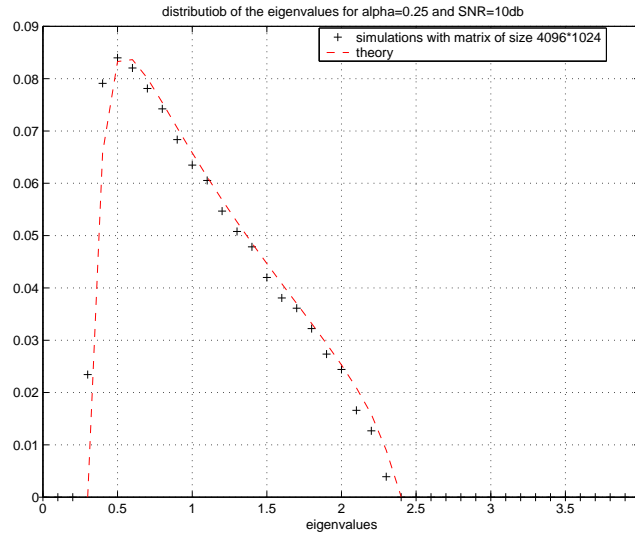


Fig. 5. matrix size: 4096*1024, 10dB, alpha=0.25.

Eq. (31) covers the cases of a number of multiple access techniques, including but not limited to Code Division Multiple Access (CDMA), Orthogonal Frequency Division Multiple Access (OFDMA) and Multiple Input Multiple Output (MIMO). Under some assumptions, the capacity of the system is given by

$$\begin{aligned}
 \frac{1}{N}C &= \frac{1}{N} \log \det \left(\mathbf{I} + \frac{1}{\sigma^2} \mathbf{H}\mathbf{H}^H \right) \\
 &= \frac{1}{N} \sum_{i=1}^N \log \left(1 + \frac{1}{\sigma^2} \lambda_i(\mathbf{H}\mathbf{H}^H) \right) \\
 &= \int \log \left(1 + \frac{1}{\sigma^2} \lambda \right) \frac{1}{N} \sum_{i=1}^N \delta(\lambda - \lambda_i(\mathbf{H}\mathbf{H}^H)) d\lambda \\
 &= \int \log \left(1 + \frac{1}{\sigma^2} \lambda \right) F^{\mathbf{H}\mathbf{H}^H}(\lambda) d\lambda.
 \end{aligned}$$

Hence, as shown by the derivation above, the empirical eigenvalue distribution naturally appears in the expression of the capacity. It also enables to derive several other performance measures of interest, such as Signal to Interference plus Noise Ratio (SINR) or multi-user efficiency [2].

Unfortunately, the three laws plotted before are among the only known empirical eigenvalue distributions which have an explicit analytical expression. Generally, the limit distributions are given by an implicit equation, and can only be computed numerically. It may look tiresome to first get the Stieltjes transform, and then retrieve the empirical eigenvalue distribution using the inversion formula. Fortunately, there is a way to circumvent this problem. One can observe that:

$$\frac{1}{N}C = \int \log \left(1 + \frac{1}{\sigma^2} \lambda \right) F^{\mathbf{H}\mathbf{H}^H}(\lambda) d\lambda,$$

and differentiating according to σ^2 , we obtain

$$\begin{aligned} \frac{1}{N} \frac{\partial C}{\partial \sigma^2} &= \int \frac{-\frac{1}{\sigma^4} \lambda}{1 + \frac{1}{\sigma^2} \lambda} F^{\mathbf{H}\mathbf{H}^H}(\lambda) d\lambda \\ &= -\frac{1}{\sigma^2} \int \frac{\frac{1}{\sigma^2} \lambda + 1 - 1}{\frac{1}{\sigma^2} \lambda + 1} F^{\mathbf{H}\mathbf{H}^H}(\lambda) d\lambda \\ &= -\frac{1}{\sigma^2} + \int \frac{1}{\lambda + \sigma^2} F^{\mathbf{H}\mathbf{H}^H}(\lambda) d\lambda \\ &= -\frac{1}{\sigma^2} + m^{\mathbf{H}\mathbf{H}^H}(-\sigma^2). \end{aligned}$$

Hence, finding the Stieltjes transform is often enough.

References

1. D.N.C Tse and S. Hanly, "Linear Multi-user Receiver: Effective Interference, Effective Bandwidth and User Capacity," pp. 641–657, Mar. 1999.
2. S. Verdu and S. Shamai, "Spectral Efficiency of CDMA with Random Spreading," pp. 622–640, Mar. 1999.
3. J. Evans and D.N.C Tse, "Large System Performance of Linear Multiuser Receivers in Multipath Fading Channels," pp. 2059–2078, Sept. 2000.
4. S. Shamai and S. Verdu, "The Impact of Frequency-Flat Fading on the Spectral Efficiency of CDMA," pp. 1302–1326, May 2001.
5. A. Tulino, L. Li, and S. Verdu, "Spectral Efficiency of Multicarrier CDMA," pp. 479 – 505, February 2005.
6. A. M. Tulino and S. Verdu, *Random Matrix Theory and Wireless Communications*, Foundations and Trends in Communications and Information Theory, NOW, The Essence of Knowledge, 1st edition, 2004.
7. J. Wishart, "The Generalized Product Moment Distribution in Samples from a Normal Multivariate Population," *Biometrika*, vol. 20, no. A, pp. 32–52, 1928.
8. E. Wigner, "Characteristic Vectors of Bordered Matrices with Infinite Dimensions," *Annals of Mathematics*, vol. 62, pp. 546–564, 1955.
9. V.A. Marchenko and L.A. Pastur, "Distribution of Eigenvalues for Some Sets of Random Matrices," *Math USSR Sb.*, vol. 1, pp. 457–483, 1967.
10. L.A. Pastur, "On the Spectrum of Random Matrices," *Teoret. Mat. Fiz.*, vol. 10, pp. 67–74, 1972.
11. J.W. Silverstein and Z.D. Bai, "On the Empirical Distribution of Eigenvalues of a Class of Large Dimensional Random Matrices," *J. Multivariate Anal.*, vol. 54, no. 2, pp. 175–192, 1995.
12. V. L. Girko, "Theory of Random Determinants," *Kluwer Academic Publishers, Dordrecht, The Netherlands*, 1990.
13. S. Haykin, "Cognitive Radio: Brain Empowered Wireless Communications," *Journal on Selected Areas in Communications*, vol. 23, pp. 201–220, 2005.
14. F. Hiai and D. Petz, *The Semicircle Law, Free Random Variables and Entropy*, American Mathematical Society, 2000.
15. D.V. Voiculescu, "Addition of certain non-commuting random variables," *J. Funct. Anal.*, vol. 66, pp. 323–335, 1986.
16. D. V. Voiculescu, "Multiplication of certain noncommuting random variables," *J. Operator Theory*, vol. 18, no. 2, pp. 223–235, 1987.
17. D.V. Voiculescu, "Circular and semicircular systems and free product factors," vol. 92, 1990.
18. D.V. Voiculescu, "Limit laws for random matrices and free products," *Inv. Math.*, vol. 104, pp. 201–220, 1991.
19. Ø. Ryan and M. Debbah, "Free deconvolution for signal processing applications," *second round review, IEEE Trans. on Information Theory*, 2008, <http://arxiv.org/abs/cs.IT/0701025>.
20. Ø. Ryan and M. Debbah, "Channel capacity estimation using free probability theory," *to appear*, 2008, <http://arxiv.org/abs/0707.3095>.
21. Ø. Ryan and M. Debbah, "Multiplicative free convolution and information-plus-noise type matrices," *Planned for submission to Journal Of Functional Analysis*, 2006, <http://www.ifi.uio.no/~oyvindry/multfreeconv.pdf>.
22. O. Ryan, "Implementation of free deconvolution," *Planned for submission to IEEE Transactions on Signal Processing*, 2006, <http://www.ifi.uio.no/~oyvindry/freedeconvsigprocessing.pdf>.
23. B. Dozier and J.W. Silverstein, "On the empirical distribution of eigenvalues of large dimensional information-plus-noise type matrices," *Submitted.*, 2004, <http://www4.ncsu.edu/~jack/infnoise.pdf>.
24. Florent Benaych-Georges, "Rectangular random matrices, related convolution," *To appear in Probability Theory and Related Fields*.
25. Benaych-Georges Florent Guionnet Alice Belinschi, Serban, "Regularization by free additive convolution, square and rectangular cases," *To appear in Complex Analysis and Operator Theory*.
26. Florent Benaych-Georges, "Infinitely divisible distributions for rectangular free convolution: classification and matricial interpretation," *Probab. Theory Related Fields*, vol. 139, no. 1-2, pp. 143–189, 2007.
27. D. V. Voiculescu, K. J. Dykema, and A. Nica, *Free random variables*, vol. 1 of *CRM Monograph Series*, American Mathematical Society, Providence, RI, 1992, A noncommutative probability approach to free products with applications to random matrices, operator algebras and harmonic analysis on free groups.

28. T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, 1991.
29. D. Guo, S. Shamai, and S. Verdú, “Mutual information and minimum mean-square error in gaussian channels,” vol. 51, no. 4, pp. 1261 – 1282, Apr. 2005.
30. D. P. Palomar and S. Verdú, “Gradient of Mutual Information in Linear Vector Gaussian Channels,” *IEEE Trans. Information Theory*, vol. 52, pp. 141–154, 2006.
31. R. Seroul and D. O’Shea, *Programming for Mathematicians*, Springer.
32. R. Couillet and M. Debbah, “Free deconvolution for ofdm multicell snr detection,” *PIMRC 2008, Cognitive Radio Workshop, Cannes, France*, Sept, 2008.
33. N.R. Rao and A. Edelman, “Free probability, sample covariance matrices and signal processing,” *ICASSP*, pp. 1001–1004, 2006.
34. N. Rao, J. Mingo, R. Speicher, and A. Edelman, “Statistical eigen-inference from large wishart matrices,” <http://front.math.ucdavis.edu/0701.5314>, Jan, 2007.
35. N. El. Karoui, “Spectrum estimation for large dimensional covariance matrices using random matrix theory,” <http://front.math.ucdavis.edu/0609.5418>, Sept, 2006.
36. E. Wigner, “On the distribution of roots of certain symmetric matrices,” *The Annals of Mathematics*, vol. 67, pp. 325–327, 1958.
37. V. L. Girko, “Circular Law,” *Theory. Prob. Appl.*, pp. 694–706, vol. 29 1984.
38. Z. D. Bai, “The Circle Law,” *The Annals of Probability.*, pp. 494–529, 1997.
39. A.M. Tulino and S. Verdo, *Random Matrix Theory and Wireless Communications*, www.nowpublishers.com, 2004.
40. Alexandru Nica and Roland Speicher, *Lectures on the combinatorics of free probability*, vol. 335 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, Cambridge, 2006.
41. Florent Benaych-Georges, “Infinitely divisible distributions for rectangular free convolution: classification and matricial interpretation,” *Probab. Theory Related Fields*, vol. 139, no. 1-2, pp. 143–189, 2007.
42. E. Telatar, “Capacity of Multi-Antenna Gaussian Channels,” *Eur. Trans. Telecomm. ETT*, vol. 10, no. 6, pp. 585–596, Nov. 1999.
43. D. Tse and S. Hanly, “Linear multiuser receivers: Effective interference, effective bandwidth and user capacity,” *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 641–657, 1999.
44. V.A. Marchenko and L.A. Pastur, “Distribution of Eigenvalues for Some Sets of Random Matrices,” *Math USSR Sb.*, vol. 1, pp. 457–483, 1967.

Tools from Physics and Road-traffic Engineering for Dense Ad-hoc Networks

Eitan Altman¹, Pierre Bernhard², Mérouane Debbah³, Alonso Silva¹

¹ Institut National de Recherche en Informatique et Automatique,
INRIA Sophia Antipolis,
F-06902 Sophia Antipolis, France

{eitan.altman, alonso.silva}@sophia.inria.fr

² I3S, University of Nice-Sophia Antipolis and CNRS
F-06902 Sophia Antipolis, France
France

Pierre.Bernhard@polytech.unice.fr

³ Supélec, Ecole supérieure d'électricité
91192 - Gif sur Yvette Cedex, France
merouane.debbah@supelec.fr

Abstract. We consider massively dense ad-hoc networks and study their continuum limits as the node density increases and as the graph providing the available routes becomes a continuous area with location- and congestion-dependent costs. We study both the global optimal solution as well as the non-cooperative routing problem among a large population of users where each user seeks a path from its source to its destination so as to minimise its individual cost. We seek a (continuum version of the) Wardrop equilibrium. We first show how to derive meaningful cost models as a function of the scaling properties of the capacity of the network and of the density of nodes. We present various solution methodologies for the problem: (1) the viscosity solution of the Hamilton-Bellman-Jacobi equation, for the global optimisation problem, (2) a method based on Green's Theorem for the least cost problem of an individual, and (3) a solution of the Wardrop equilibrium problem using a transformation into an equivalent global optimisation problem.

1 Introduction

In the design and analysis of wireless networks, researchers frequently stumble on the scalability problem that can be summarised in the following sentence: “As the number of nodes in the network increases, problems become harder to solve” [26]. The sentence takes its meaning from several issues. Some examples are the following:

- In Routing: As the network size increases, routes consists of an increasing number of nodes, and so they are increasingly susceptible to node mobility and channel fading [22].
- In Transmission Scheduling: The determination of the maximum number of non-conflicting transmissions in a graph is a NP-complete problem [29].
- In Capacity of Wireless Networks: As the number of nodes increases, the determination of the precise capacity becomes an intractable problem.

Nevertheless when the system is sufficiently large, one may hope that a macroscopic model will give a better description of the network and that one could predict its properties from microscopic considerations. Indeed we are going to sacrifice some details, but this macroscopic view will preserve sufficient information to allow a meaningful network optimisation solution and the derivation of insightful results in a wide range of settings.

The physics-inspired paradigms used for the study of large ad-hoc networks go way beyond those related to statistical-mechanics in which macroscopic properties are derived from microscopic structure. Starting from the pioneering work by Jacquet (see [17]) in that area, a number of research groups have worked on massively dense ad-hoc networks using tools from geometrical optics [17]⁴ as well as electrostatics (see e.g. [26,25,13], and the survey [27] and references therein). We shall describe these in the next sections.

⁴ We note that this approach is restricted to costs that do not depend on the congestion

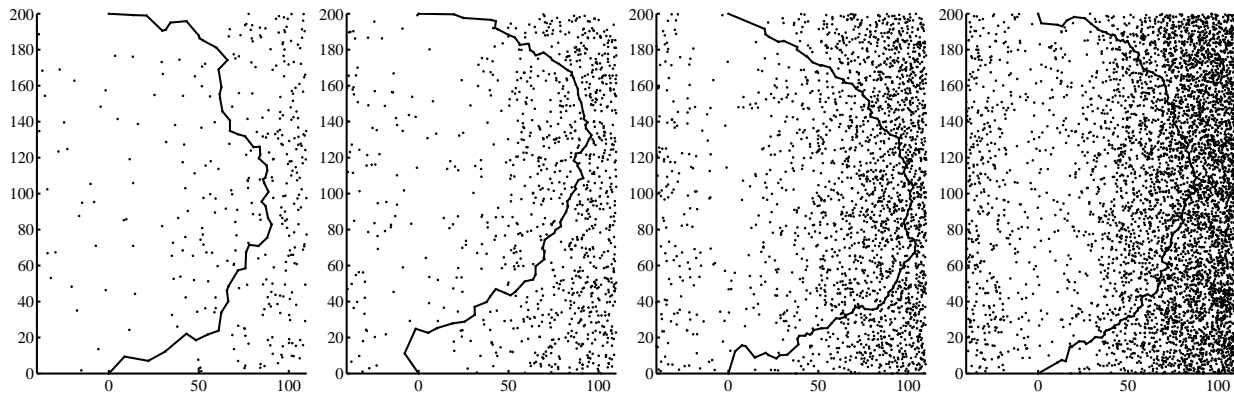


Fig. 1. Minimum cost routes in increasingly large networks.

The physical paradigms allow the authors to minimise various metrics related to the routing. In contrast, Hyytia and Virtamo propose in [15] an approach based on load balancing arguing that if shortest path (or cost minimisation) arguments were used, then some parts of the network would carry more traffic than others and may use more energy than others. This would result in a shorter lifetime of the network since some parts would be out of energy earlier than others and earlier than any part in a load balanced network.

The term “massively dense” ad-hoc networks is used to indicate not only that the number of nodes is large, but also that the network is highly connected. By the term “dense” we further understand that for every point in the plain there is a node close to it with high probability ; by “close” we mean that its distance is much smaller than the transmission range. In this chapter and in the work (cited in the next paragraphs) one actually studies the limiting properties of massively dense ad-hoc networks, as the density of nodes tends to infinity. The existence of such a limit is illustrated in Fig. 1 that was kindly made available to us by Toumpis. In the figure, which appeared in [27], the author plots the minimum-cost route connecting a source node placed at the origin (0 m,0 m) and a destination node placed at the location (0 m,200 m), through an area where relay nodes are placed according to a spatial Poisson process of density $\lambda(\mathbf{x}) = a \cdot [10^{-4}x_1^2 + 0.05]$ nodes per m^2 , for four increasing values of a ($a = \frac{1}{30}, \frac{1}{10}, \frac{1}{5}, \frac{1}{2}$). The author makes in the figures the assumption that the cost connecting two nodes that are separated by a distance δ equals $c(\delta) = \delta^2$. From the figures we can see that as the number of nodes increases, the optimal route starts more and more to resemble a continuous curve, and it turns out that the shape of the curve does not depend on the precise node placement, but only on the node density function $\lambda(\mathbf{x})$ and the cost-versus-distance function $c(\delta)$.

The development of the original theory of routing in massively dense networks among the community of ad-hoc networks has emerged in a complete independent way of the existing theory of routing in massively dense networks which had been developed within the community of road traffic engineers. Indeed, this approach had already been introduced in 1952 by Wardrop [30] and by Beckmann [4] and is still an active research area among that community, see [6,7,14,16,32] and references therein. We combine in this chapter various approaches from this area as well as from optimal control theory in order to formulate models for routing in massively dense networks. We further propose simple novel approach to that problem using a classical device of 2-D. singular optimal control [19] based on Green’s formula to obtain a simple characterisation of least cost paths of individual packets. We end the chapter by a numerical example for computing an equilibrium.

We consider in this chapter static networks (say sensor networks) characterized by communications through horizontally and vertically oriented directional antennas. The use of directional antennas allows one to save energy and to use it in an efficient way which may result in a longer life time of the network.

The structure of this chapter is as follows. We begin by presenting models for costs relevant to optimisation models in routing or to node assignment. We then formulate the global optimisation problem and the individual optimisation one with a focus on the directional antennas scenario. We provide several approaches to obtain both qualitative characterisation as well as quantitative solutions to the problems.

2 Determining routing costs in dense ad-hoc networks

In optimising a routing protocol in ad-hoc networks, or in optimising the placement of nodes, one of the starting points is the determination of the cost function. To determine it, we need a detailed specification of the network which includes the following:

- A model for the placement of nodes in the network.
- A forward rule that nodes will use to select the next hop of a packet.
- A model for the cost incurred in one hop, i.e. for transmitting a packet to an intermediate node.

Below we present several ways of choosing cost functions.

2.1 Costs derived from capacity scaling

Many models have been proposed in the literature that show how the transport capacity scales with the number of nodes n or with their density λ . Assume that we use a protocol that provides a transport capacity of the order of $f(\lambda)$ at some region in which the density of nodes is λ . A typical cost (see e.g. [25]) at a neighbourhood of \mathbf{x} is the density of nodes required there to carry a given flow. Assuming that a flow⁵ $\mathbf{T}(\mathbf{x})$ is assigned through a neighbourhood of \mathbf{x} , the cost is taken to be

$$c(\mathbf{x}, \mathbf{T}(\mathbf{x})) = f^{-1}(|\mathbf{T}(\mathbf{x})|) \quad (1)$$

where $|\cdot|$ represents the norm of a vector.

Examples for f :

- Using a network theoretic approach based on multi-hop communication, Gupta and Kumar prove in [12] that the throughput of the system that can be transported by the network when the nodes are optimally located is $\Omega(\sqrt{\lambda})$, and when the nodes are randomly located this throughput becomes $\Omega(\frac{\sqrt{\lambda}}{\sqrt{\log \lambda}})$. Using percolation theory, the authors of [9] have shown that in the randomly located set the same $\Omega(\sqrt{\lambda})$ can be achieved.
- Baccelli, Blaszczyzyn and Mühlethaler introduce in [2] an access scheme, MSR (Multi-hop Spatial Reuse Aloha), reaching the Gupta and Kumar bound $O(\sqrt{\lambda})$ which does not require prior knowledge of the node density.
- A protocol introduced by Tse and Glosglauer [10] has a capacity that scales as $O(\lambda)$. However, it does not fall directly within the class of massively dense ad-hoc networks and indeed, it relies on mobility and on relaying for handling disconnectivity.

We conclude that for the model of Gupta and Kumar with either the optimal location or the random location approaches, as well as for the MSR protocol with a Poisson distribution of nodes, we obtain a quadratic cost of the form

$$c(\mathbf{T}(\mathbf{x})) = k|\mathbf{T}(\mathbf{x})|^2 \quad (2)$$

This follows from (1) as $f(x)$ behaves like \sqrt{x} so its inverse is quadratic.

2.2 Congestion-independent routing

A metric often used in the Internet for determining routing is the number of hops, which routing protocol try to minimise. The number of hops is proportional to the expected delay along the path in the context of ad-hoc networks, in case that the queueing delay is negligible with respect to the transmission delay over each hop. This criterion is insensitive to interference or congestion. We assume that it depends only on the transmission range. We describe various cost criteria that can be formulated with this approach.

- If the range were constant then the cost density $c(\mathbf{x})$ is constant so that the cost of a path is its length in meters. The routing then follows a shortest path selection.

⁵ We denote with bold font the vectors.

- Let us assume that the range $R(\mathbf{x})$ depends on local radio conditions at a point \mathbf{x} (for example, if it is influenced by weather conditions) but not on interference. The latter is justified when dedicated orthogonal channels (e.g. in time or frequency) can be allocated to traffic flows that would otherwise interfere with each other. Then determining the routing becomes a path cost minimisation problem. We further assume, as in Gupta and Kumar, that the range is scaled to go to 0 as the total density λ of nodes grows to infinity. More precisely, let us consider a scaling of the range such that the following limit exists:

$$r(\mathbf{x}) := \lim_{\lambda \rightarrow \infty} \frac{R^\lambda(\mathbf{x})}{\lambda}$$

Then in the dense limit, the fraction of nodes that participate in forwarding packets along a path is $1/r(\mathbf{x})$ and the path cost is the integral of this density along the path.

- The influence of varying radio conditions on the range can be eliminated using power control that can equalise the hop distance.

2.3 Costs related to energy consumption

In the absence of capacity constraints, the cost can represent energy consumption. In a general multi-hop ad-hoc network, the hop distance can be optimised so as to minimise the energy consumption. Even within a single cell of 802.11 IEEE wireless LAN one can improve the energy consumption by using multiple hops, as it has been shown not to be efficient in terms of energy consumption to use a single hop [20].

Alternatively, the cost can take into account the scaling of the nodes (as we had in Subsection 2.1) that is obtained when there are energy constraints. As an example, assuming random deployment of nodes, where each node has data to send to another randomly selected node, the capacity (in bits per Joule) has the form $f(\lambda) = \Omega((\lambda/\log \lambda)^{(q-1)/2})$ where q is the path-loss, see [21]. The cost is then obtained using (1).

3 Preliminary

In the work of Toumpis et al. ([26,25,13,28,27]), the authors address the problem of the optimal deployment of Wireless Sensor Networks by a parallel with Electrostatic.

Consider in the two dimensional plane $X_1 \times X_2$, the continuous **information density function** $\rho(\mathbf{x})$, measured in bps/m², such that at locations \mathbf{x} where $\rho(\mathbf{x}) > 0$ there is a distributed data source such that the rate with which information is created in an infinitesimal area of size $d\Omega$ centred at \mathbf{x} is $\rho(\mathbf{x})d\Omega$. Similarly, at locations \mathbf{x} where $\rho(\mathbf{x}) < 0$ there is a distributed data sink such that the rate with which information is absorbed by an infinitesimal area of size $d\Omega$, centred at point \mathbf{x} , is equal to $-\rho(\mathbf{x})d\Omega$.

The total rate at which sinks must absorb data is the same as the total rate which the data is created at the sources, i.e.

$$\int_{X_1 \times X_2} \rho(\mathbf{x})dS = 0.$$

Next we present the flow conservation condition (see e.g. [25,6] for more details). For information to be conserved over a domain Ω_0 of arbitrary shape on the $X_1 \times X_2$ plane, (but with smooth boundary) it is necessary that the rate with which information is created in the area is equal to the rate with which information is leaving the area, i.e

$$\int_{\Omega_0} \rho(\mathbf{x})d\mathbf{x} = \oint_{\partial\Omega_0} [\mathbf{T}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})]d\ell$$

The integral on the left is the surface integral of $\rho(\mathbf{x})$ over Ω_0 . The integral on the right is the path integral of the inner product $\mathbf{T} \cdot \mathbf{n}$ over the curve $\partial\Omega_0$. The vector $\mathbf{n}(\mathbf{x})$ is the unit normal vector to $\partial\Omega_0$ at the boundary point $\mathbf{x} \in \partial\Omega_0$ and pointing outwards. The function $\mathbf{T}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$ measured in bps/m² is equal at the rate with which information is leaving the domain Ω_0 per unit length of boundary at the boundary point \mathbf{x} .

This holding for *any* (smooth) domain Ω_0 , it follows that necessarily

$$\nabla \cdot \mathbf{T}(\mathbf{x}) := \frac{\partial T_1(\mathbf{x})}{\partial x_1} + \frac{\partial T_2(\mathbf{x})}{\partial x_2} = \rho(\mathbf{x}), \quad (3)$$

where “ $\nabla \cdot$ ” is the divergence operator.

Extension to multi-class The work on massively dense ad-hoc networks considered a single class of traffic. In the geometrical optics approach it corresponded to demand from a point \mathbf{a} to a point \mathbf{b} . In the electrostatic case it corresponded to a set of origins and a set of destinations where traffic from any origin point could go to any destination point. The analogy to positive and negative charges in electrostatics may limit the perspectives of multi-class problems where traffic from distinct origin sets has to be routed to distinct destination sets.

The model based on geometrical optics can directly be extended to include multiple classes as there are no elements in the model that suggest coupling between classes. This is due in particular to the fact that the cost density has been assumed to depend only on the density of the mobiles and not on the density of the flows.

In contrast, the cost in the model based on electrostatics is assumed to depend both on the location as well as on the local flow density. It thus models more complex interactions that would occur if we considered the case of ν traffic classes. Extending the relation (3) to the multi-class case, we have traffic conservation at each point in space to each traffic class as expressed in the following:

$$\nabla \cdot \mathbf{T}^j(\mathbf{x}) = \rho^j(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \quad (4)$$

The function \mathbf{T}^j is the flow distribution of class j and ρ^j corresponds to the distribution of the external sources and/or sinks.

Let $\mathbf{T}(\mathbf{x})$ be the total flow vector at point $\mathbf{x} \in \Omega$. A generic multi-class optimisation problem would then be: minimise Z over the flow distributions $\{T_i^j\}$

$$Z = \int_{\Omega} g(\mathbf{x}, \mathbf{T}(\mathbf{x})) dx_1 dx_2 \quad \text{subject to} \quad \nabla \cdot \mathbf{T}^j(\mathbf{x}) = \rho^j(\mathbf{x}), \quad j = 1, \dots, \nu \quad \forall \mathbf{x} \in \Omega. \quad (5)$$

4 Directional Antennas and Global Optimisation

Unlike the previous work that we described on massively dense ad-hoc networks, we introduce a model that uses directional antennas. The approach that we follow is inspired by the work of Dafermos (see [6]) on road traffic. An alternative approach based on road traffic tools can be found in [1,23].

For energy efficiency, it is assumed that each terminal is equipped with one or with two directional antennas, allowing transmission at each hop to be directed either from North to South or from West to East. The model we use extends that of [6] to the multi-class framework. We thus consider ν classes of flows $T_1^j \geq 0$, $T_2^j \geq 0$, $j = 1, \dots, \nu$. To be compatible with Dafermos [6], we use her definitions of orientation according to which the directions North to South and West to East are taken positive. In the dense limit, a curved path can be viewed as a limit of a path with many such hops as the hop distance tends to zero.

Some assumptions on the cost:

- **Individual cost:** We allow the cost for a horizontal (West-East) transmission from a point \mathbf{x} to be different than the cost for a vertical transmission (North-South). It is assumed that a packet located at the point \mathbf{x} and travelling in the direction of the axis x_i incurs a **transportation cost** g_i and such transportation cost depends upon the position \mathbf{x} and the traffic flow $\mathbf{T}(\mathbf{x})$. We thus allow for a vector valued cost $\mathbf{g} := \mathbf{g}(\mathbf{x}, \mathbf{T}(\mathbf{x}))$.
- The local cost corresponding to the global optimisation problem is given by $g(\mathbf{x}, \mathbf{T}(\mathbf{x})) = \mathbf{g}(\mathbf{x}, \mathbf{T}(\mathbf{x})) \cdot \mathbf{T}(\mathbf{x})$ if it is perceived as the sum of costs of individuals.
- The global cost will be the integral of the local cost density.
- The local cost $g(\mathbf{x}, \mathbf{T}(\mathbf{x}))$ is assumed to be non-negative, convex increasing in each of the components of \mathbf{T} (T_1 and T_2 in our 2-dimensional case).

The **boundary conditions** will be determined by the options that travellers have in selecting their origins and/or destinations. Examples of the boundary conditions are:

- *Assignment problem*: users of the network have predetermined origins and destinations and are free to choose their travel paths.
- *Combined distribution and assignment problem*: users of the network have predetermined origins and are free to choose their destinations (within a certain destination region) as well as their paths.
- *Combined generation, distributions and assignment problem*: users are free to choose their origins, their destinations, as well as their travel paths.

The problem formulation is again to minimise Z as defined in (5). The natural choice of functional spaces to make that problem precise, and take advantage of the large body of theory developed with Sobolev spaces in the PDE community, is to seek T_i^j in $L^2(\Omega)$, so that ρ may be in $H^{-1}(\Omega)$, allowing for some localised mass of traffic source or sink.

Kuhn-Tucker conditions. Define the Lagrangian as

$$L^\zeta(\mathbf{x}, \mathbf{T}) := \int_{\Omega} \ell^\zeta(\mathbf{x}, \mathbf{T}) \, d\mathbf{x} \quad \text{where } \ell^\zeta(\mathbf{x}, \mathbf{T}) = g(\mathbf{x}, \mathbf{T}(\mathbf{x})) - \sum_{j=1}^v \zeta^j(\mathbf{x}) \left[\nabla \cdot \mathbf{T}^j(\mathbf{x}) - \rho^j(\mathbf{x}) \right]$$

where the $\zeta^j(\mathbf{x}) \in H^1(\Omega)$ are Lagrange multipliers. The criterion is convex, and the constraint (4) affine. Therefore the Kuhn-Tucker theorem holds, stating that the Lagrangian is minimum at the optimum. A variation $\delta\mathbf{T}(\cdot)$ will be admissible if $\mathbf{T}(\mathbf{x}) + \delta\mathbf{T}(\mathbf{x}) \geq 0$ for all \mathbf{x} , hence in particular, $\forall \mathbf{x} : T_i^j(\mathbf{x}) = 0, \delta T_i^j(\mathbf{x}) \geq 0$.

Let DL^ζ denote the Gâteaux derivative of L^ζ w.r. to $T(\cdot)$. Euler’s inequality reads

$$\forall \delta\mathbf{T} \text{ admissible}, DL^\zeta \cdot \delta\mathbf{T} \geq 0,$$

therefore here

$$\int_{\Omega} \sum_j \langle \nabla_{\mathbf{T}^j} g(\mathbf{x}, \mathbf{T}(\mathbf{x})), \delta\mathbf{T}^j(\mathbf{x}) \rangle \, d\mathbf{x} - \int_{\Omega} \sum_j \zeta^j(\mathbf{x}) \nabla \cdot \delta\mathbf{T}^j(\mathbf{x}) \, d\mathbf{x} \geq 0.$$

Integrating by parts with Green’s formula, this is equivalent to

$$\int_{\Omega} \sum_j [\langle \nabla_{\mathbf{T}^j} g, \delta\mathbf{T}^j \rangle + \langle \nabla_{\mathbf{x}} \zeta^j, \delta\mathbf{T}^j \rangle] \, d\mathbf{x} - \int_{\partial\Omega} \sum_j \zeta^j \langle \delta\mathbf{T}^j, \mathbf{n} \rangle \, d\ell \geq 0.$$

We may choose all the $\delta\mathbf{T}^k = 0$ except $\delta\mathbf{T}^j$, and choose that one in $(H_0^1(\Omega))^2$, *i.e.* such that the boundary integral be zero. This is always feasible and admissible. Then the last term above vanishes, and it is a classical fact that the inequality implies for $i = 1, 2$:

$$\frac{\partial g(\mathbf{x}, \mathbf{T})}{\partial T_i^j} + \frac{\partial \zeta^j(\mathbf{x})}{\partial x_i} = 0 \quad \text{if } T_i^j(\mathbf{x}) > 0 \tag{6a}$$

$$\frac{\partial g(\mathbf{x}, \mathbf{T})}{\partial T_i^j} + \frac{\partial \zeta^j(\mathbf{x})}{\partial x_i} \geq 0 \quad \text{if } T_i^j(\mathbf{x}) = 0. \tag{6b}$$

Placing this back in Euler’s inequality, and using a $\delta\mathbf{T}^j$ non zero on the boundary, it follows that necessarily $\zeta^j(\mathbf{x}) = 0$ at any \mathbf{x} of the boundary $\partial\Omega$ where $T(\mathbf{x}) > 0$. This provides the boundary condition to recover ζ^j from the condition (4).

Remark: The Kuhn-Tucker type characterisation (6a)-(6b) is already stated in [6] for the single class case. However, as Dafermos states explicitly, its rigorous derivation is not available there.

Consider the following special cases that we shall need later. We assume a single traffic class, but this could easily be extended to several. Let

$$g(\mathbf{x}, \mathbf{T}(\mathbf{x})) = \sum_{i=1,2} g_i(\mathbf{x}, \mathbf{T}(\mathbf{x})) T_i(\mathbf{x}).$$

⁶ This is a complementary slackness condition on the boundary.

1. Monomial cost per packet:

$$g_i(\mathbf{x}, \mathbf{T}(\mathbf{x})) = k_i(\mathbf{x}) \left(T_i(\mathbf{x}) \right)^\beta \quad (7)$$

for some $\beta > 1$. Then (6a)-(6b) simplify to

$$(\beta + 1)k_i(\mathbf{x}) (T_i(\mathbf{x}))^\beta + \frac{\partial \zeta(\mathbf{x})}{\partial x_i} = 0 \quad \text{if } T_i(\mathbf{x}) > 0 \quad (8a)$$

$$(\beta + 1)k_i(\mathbf{x}) (T_i(\mathbf{x}))^\beta + \frac{\partial \zeta(\mathbf{x})}{\partial x_i} \geq 0 \quad \text{if } T_i(\mathbf{x}) = 0. \quad (8b)$$

In that case, recovery of ζ to complete the process is difficult, at best. Things are simpler in the next case.

2. Affine cost per packet:

$$g_i(\mathbf{x}, \mathbf{T}(\mathbf{x})) = \frac{1}{2}k_i(\mathbf{x})T_i(\mathbf{x}) + h_i(\mathbf{x}). \quad (9)$$

Then (6a)-(6b) simplify to

$$k_i(\mathbf{x})T_i(\mathbf{x}) + h_i(\mathbf{x}) + \frac{\partial \zeta(\mathbf{x})}{\partial x_i} = 0 \quad \text{if } T_i(\mathbf{x}) > 0$$

$$k_i(\mathbf{x})T_i(\mathbf{x}) + h_i(\mathbf{x}) + \frac{\partial \zeta(\mathbf{x})}{\partial x_i} \geq 0 \quad \text{if } T_i(\mathbf{x}) = 0.$$

Assume that the $k_i(\cdot)$ are everywhere positive and bounded away from 0. For simplicity, let $a_i = 1/k_i$, and b be the vector with coordinates $b_i = h_i/k_i$, all assumed to be square integrable. Assume that there exists a solution where $T(\mathbf{x}) > 0$ for all \mathbf{x} . Then

$$T_i(\mathbf{x}) = - \left(a_i(\mathbf{x}) \frac{\partial \zeta(\mathbf{x})}{\partial x_i} + b_i(\mathbf{x}) \right).$$

As a consequence, from (4) and the above remark, we get that $\zeta(\cdot)$ is to be found as the solution in $H_0^1(\Omega)$ of the elliptic equation (an equality in $H^{-1}(\Omega)$)

$$\sum_i \frac{\partial}{\partial x_i} \left(a_i(\mathbf{x}) \frac{\partial \zeta}{\partial x_i} \right) + \nabla \cdot b(\mathbf{x}) + \rho(\mathbf{x}) = 0.$$

This is a well behaved Dirichlet problem, known to have a unique solution in $H_0^1(\Omega)$, furthermore easy to compute numerically.

5 User optimisation and congestion independent costs

We expand on the shortest path approach for optimisation that has already appeared using geometrical optics tools [17]. We present general optimisation frameworks for handling shortest path problems and more generally, minimum cost paths.

We consider the model of Section 4. We assume that the local cost depends on the direction of the flow but not on its size. The cost is $c_1(\mathbf{x})$ for a flow that is locally horizontal and is $c_2(\mathbf{x})$ for a flow that is locally vertical. We assume in this section that c_1 and c_2 do not depend on \mathbf{T} . The cost incurred by a packet transmitted along a path p is given by the line integral

$$\mathbf{c}_p = \int_p \mathbf{c} \cdot d\mathbf{x}. \quad (11)$$

Let $V^j(\mathbf{x})$ be the minimum cost to go from a point \mathbf{x} to a set B^j , $j = 1, \dots, v$. Then

$$V^j(\mathbf{x}) = \min (c_1(\mathbf{x})dx_1 + V^j(x_1 + dx_1, x_2), c_2(\mathbf{x})dx_2 + V^j(x_1, x_2 + dx_2)) \quad (12)$$

This can be written as

$$0 = \min \left(c_1(\mathbf{x}) + \frac{\partial V^j(\mathbf{x})}{\partial x_1}, c_2(\mathbf{x}) + \frac{\partial V^j(\mathbf{x})}{\partial x_2} \right), \quad \forall \mathbf{x} \in B^j, V^j(\mathbf{x}) = 0. \quad (13)$$

If V^j is differentiable then, under suitable conditions, it is the unique solution of (13). In the case that V^j is not everywhere differentiable then, under suitable conditions, it is the unique viscosity solution of (13) (see [3,8]).

There are many numerical approaches for solving the HJB equation. One can discretise the HJB equation and obtain a discrete dynamic programming for which efficient solution methods exist. If one repeats this for various discretisation steps, then we know that the solution of the discrete problem converges to the viscosity solution of the original problem (under suitable conditions) as the step size converges to zero [3].

6 Geometry of minimum cost paths

We begin by introducing the standard attribute (plus or minus) to a path according to the direction of the movement along it. The definition is different than in [6] (which we used in Section 4).

Definition 1. [18]. (i) Let C be some simple closed curve surrounding some region R . Then C^+ corresponds to a counterclockwise movement; more precisely, it corresponds to moving so that the region R is to our left. The opposite orientation along C is denoted by C^- .

(ii) The orientation of path segments which are not closed are defined differently. A “plus” indicates an orientation of left to right or bottom to top, and the “minus” indicates curves oriented from right to left or from top to bottom.

We consider now our directional antenna model in a given rectangular area R on a region Ω , defined by the simple closed curve $\partial R^+ = \Gamma_1^+ \cup \Gamma_2^+ \cup \Gamma_3^- \cup \Gamma_4^-$ (see Fig. 2).

We obtain below **optimal paths** defined as paths that achieve the minimum cost in (11). We shall study two problems:

- **Point to point optimal path:** we seek the minimum cost path between two points.
- **Point to boundary optimal path:** we seek the minimum cost path on a given region that starts at a given point and is allowed to end at any point on the boundaries.

Define the function

$$U(\mathbf{x}) = \frac{\partial c_2}{\partial x_1}(\mathbf{x}) - \frac{\partial c_1}{\partial x_2}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega.$$

It will turn out that the structure of the minimum cost path depends on the costs through the sign of the function U . Now, if the function $\mathbf{c} \in C^1(\Omega)$ then U is a continuous function on Ω . This motivates us to study cases in which U has the same sign everywhere (see Fig. 3), or in which there are two regions in R , one with $U > 0$ and one with $U < 0$, separated by a curve on which $U = 0$ (e.g. Fig. 4).

We shall assume throughout that the function $\mathbf{c} \in C^1(\Omega)$, and that, if non empty, the set $M = \{\mathbf{x} \mid U(\mathbf{x}) = 0\}$ is a smooth line. (This is true, e.g., if $\mathbf{c} \in C^2$ and $\nabla U \neq 0$ on M .)

6.1 The function U has the same sign over the whole region R

Theorem 1. (Point to point optimal path) Suppose that a point $\mathbf{x}^O = (x_1^O, x_2^O)$ in \mathring{R} (the interior of R), wants to send a packet to a point $\mathbf{x}^D = (x_1^D, x_2^D)$ in \mathring{R} .

- If $U > 0$ in the region $R_{OD} = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^D, x_2^O \leq x_2 \leq x_2^D\}$, except perhaps from a set of Lebesgue measure zero, then there is an optimal path given by (see Fig. 5):

$$\gamma^{opt} = \gamma_H \cup \gamma_V \text{ where}$$

$$\gamma_H = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^D, x_2 = x_2^O\}$$

$$\gamma_V = \{(x_1, x_2) \text{ such that } x_1 = x_1^D, x_2^O \leq x_2 \leq x_2^D\}.$$

$$\begin{aligned}\Gamma_1^+ &= \{0 \leq x_1 \leq a, \quad x_2 = 0\} \\ \Gamma_2^+ &= \{x_1 = a, \quad 0 \leq x_2 \leq b\} \\ \Gamma_3^- &= \{0 \leq x_1 \leq a, \quad x_2 = b\} \\ \Gamma_4^- &= \{x_1 = 0, \quad 0 \leq x_2 \leq b\}.\end{aligned}$$

Fig. 2. The boundaries of the region R .

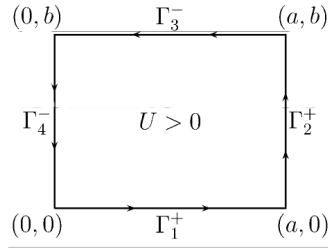


Fig. 3. The region R . The case where $U > 0$.

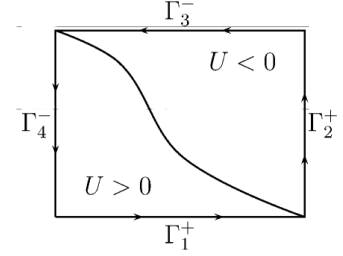


Fig. 4. The case of two regions separated by a curve. Case 1.

ii. If $U < 0$ in that region except perhaps from a set of Lebesgue measure zero, then there is an optimal path given by (see Fig. 6):

$$\gamma^{opt} = \gamma_V \cup \gamma_H \text{ where}$$

$$\gamma_V = \{(x_1, x_2) \text{ such that } x_1 = x_1^O, x_2^O \leq x_2 \leq x_2^D\}$$

$$\gamma_H = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^D, x_2 = x_2^D\}.$$

iii. In both cases, γ^{opt} is unique up to a zero Lebesgue measure. (i.e. the Lebesgue measure of the area between γ^{opt} and any other optimal path is zero).

Proof.- Consider an arbitrary path⁷ γ_C joining \mathbf{x}^O to \mathbf{x}^D , and assume that the Lebesgue measure of the area between γ^{opt} and γ_C is nonzero. We call such path, the comparison path (see Fig. 5 for the case $U > 0$ and Fig. 6 for $U < 0$).

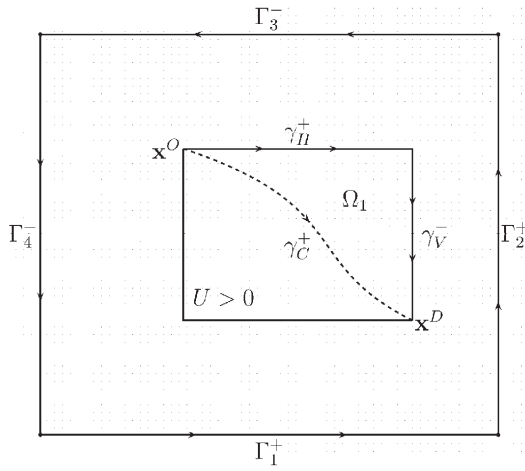


Fig. 5. Optimal path for $U > 0$.

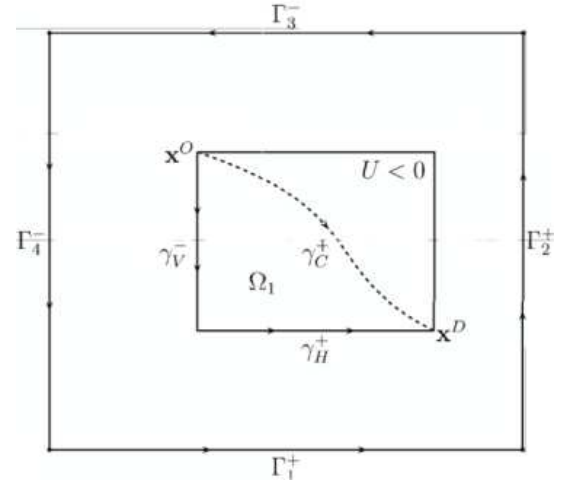


Fig. 6. Optimal path for $U < 0$.

(i) Showing that the cost over path γ^{opt} is optimal is equivalent to showing that the integral of the cost over the closed path ξ^- is negative, where ξ^- is given by following γ^{opt} from the source \mathbf{x}^O to the destination \mathbf{x}^D and then returning from \mathbf{x}^D to \mathbf{x}^O by moving along the path γ_C in the reverse direction. This closed path is written as $\xi^- = \gamma_H^+ \cup \gamma_V^- \cup \gamma_C^+$ and Ω_1 denotes the bounded area described by ξ^- . Using Green Theorem (see Appendix) we obtain

$$\oint_{\xi^-} \mathbf{c} \cdot d\mathbf{x} = - \int_{\Omega_1} U(\mathbf{x}) dS$$

⁷ Respecting that each sub-path can be decomposed in sums of paths either from North to South or from West to East (or is a limit of such paths). From now on, we will call a path valid if it satisfies that condition.

which is strictly negative since $U > 0$ a.e. on R . Decomposing the left integral, this concludes the proof of (i), and establishes at the same time the corresponding statement on uniqueness in (iii).

(ii) is obtained similarly. ■

Theorem 2. (Point to boundary optimal path)

Consider the problem of finding an optimal path from a point $\bar{\mathbf{x}} \in \mathring{R}$ to the boundary $\Gamma_1 \cup \Gamma_2$.

- i. Assume that $U(\mathbf{x}) < 0$ for all $\mathbf{x} \in \mathring{R}$ except perhaps for a set of Lebesgue measure zero. Assume that the cost on Γ_1 is non-negative and that the cost on Γ_2 is non-positive. Then the optimal path is the straight vertical line.
- ii. Assume that $U(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathring{R}$ except perhaps for a set of Lebesgue measure zero. Assume that the cost on Γ_1 is non-positive and that the cost on Γ_2 is non-negative. Then the optimal path is the straight horizontal line.

Proof.-

(i) Denote by γ_V the straight vertical path joining $\bar{\mathbf{x}}$ to Γ_1 . Consider another arbitrary valid path γ_C joining $\bar{\mathbf{x}}$ to any point \mathbf{x}^* on $\Gamma_1 \cup \Gamma_2$, and assume that the Lebesgue measure of the area between γ^{opt} and γ_C is nonzero. We call such path, the comparison path.

Assume first that \mathbf{x}^* is on Γ_2 . Denote $\mathbf{x}^D := \Gamma_1 \cap \Gamma_2$. Then by Theorem 1 (ii), the cost to go from $\bar{\mathbf{x}}$ to \mathbf{x}^D is smaller when using γ_V^- and then continuing east-wards (along Γ_1^+) than when using γ_C^+ and then south-wards (along Γ_2^*). Due to our assumptions on the costs over the boundaries, this implies that the cost along γ_V is smaller than along γ_C .

Next consider the case where \mathbf{x}^* is on Γ_1 . Denote by η the section of the boundary Γ_1 that joins $\gamma_V \cap \Gamma_1$ with \mathbf{x}^* (see Figure 7). Then again, by Theorem 1 (ii), the cost to go from $\bar{\mathbf{x}}$ to \mathbf{x}^* is smaller when using γ_V^- and then continuing east-wards (along Γ_1^+) than when using γ_C^+ . Due to our assumptions that the cost on Γ_1 is non-negative, this implies that the cost along γ_V is smaller than along γ_C .

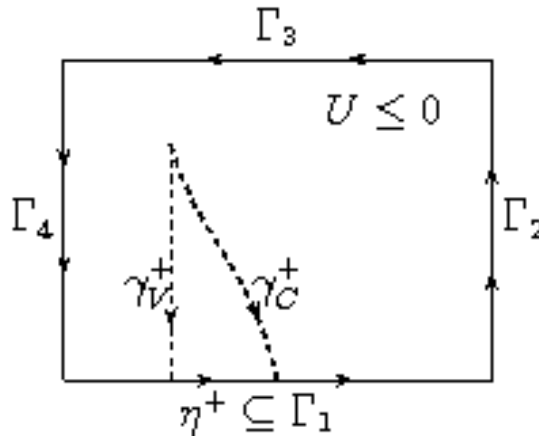


Fig. 7. Theorem 2 (i)

(ii) is obtained similarly. ■

6.2 The function U changes sign within the region R

Consider the region on the space $M := \{\mathbf{x} \in \Omega \text{ such that } U(\mathbf{x}) = 0\}$. Let us consider the case when M is only a valid path in the rectangular area, such that it starts at the intersection $\Gamma_3 \cap \Gamma_4$, and finishes at the intersection of the sinks $\Gamma_1 \cap \Gamma_2$. Then the space is divided in two areas, and as the function U is continuous we have the following cases:

- 1. $U(\mathbf{x})$ is negative in the upper area and positive in the lower area (see Fig. 4).
- 2. $U(\mathbf{x})$ is positive in the upper area and negative in the lower area (see Fig. 8).

Two other cases where the sign of U is the same over Ω are contained in what we solved in the previous section (allowing U to be zero on M which has Lebesgue measure zero)

Case 1: The function $U(\mathbf{x})$ is negative in the upper area and positive in the lower area.

We shall show that this case, M is an attractor.

Proposition 1. *Assume that the source \mathbf{x} and destination \mathbf{y} are both on M . Then the path p_M that follows M is optimal.*

Proof.- Consider an alternative path γ_C that coincides with M only in the source and destination points. First assume γ_C is entirely in the upper (i.e. northern) part and call Ω_1 the surrounded area. Define ξ^+ to be the closed path that follows p_M from \mathbf{x} to \mathbf{y} and then returns along γ_C .

The integral $\int_{\Omega_1} U(\mathbf{x})dS$ is negative by assumption. By Green Theorem it equals $\oint_{\xi^+} \mathbf{c} \cdot d\mathbf{x}$. This implies that the cost along p_M is strictly smaller than along γ_C .

A similar argument holds for the case that γ_C is below p_M .

A path between \mathbf{x} and \mathbf{y} may have several intersections with M . Between each pair of consecutive intersections of M , the sub-path has a cost larger than that obtained by following M between these points (this follows from the previous steps of the proof). We conclude that p_M is indeed optimal. ■

Proposition 2. *Let a point $\bar{\mathbf{x}}^O$ send packets to a point \mathbf{x}^D .*

- i. *Assume both points in the upper region. Denote by γ_1 the two segments path given in Theorem 1 (ii). Then the curve $\hat{\gamma}$ obtained as the maximum between M and γ_1 is optimal.⁸*
- ii. *Let both points be in the lower region. Denote by γ_2 the two segments path given in Theorem 1 (i). Then the curve $\bar{\gamma}$ obtained as the minimum between M and γ_2 is optimal.*

Proof.- (i) A straightforward adaptation of the proof of the previous proposition implies that the path in the statement of the proposition is optimal among all those restricted to the upper region. Consider now a path γ_C that is not restricted to the upper region. Then $M \cap \gamma_C$ contains two distinct points such that γ_C is strictly lower than M between these points. Applying Proposition 1 we then see that the cost of γ_C can be strictly improved by following M between these points instead of following γ_C there. This concludes (i). (ii) is proved similarly. ■

Proposition 3. *Let a point $\bar{\mathbf{x}}^O$ send packets to a point \mathbf{x}^D .*

- i. *Assume the origin is in the upper region and the destination in the lower one. Then the optimal path has three segments;*
 1. *It goes straight vertically from $\bar{\mathbf{x}}^O$ to M ,*
 2. *Continues as long as possible along M , i.e. until it reaches the x coordinate of the destination,*
 3. *At that point it goes straight vertically from M to \mathbf{x}^D .*
- ii. *Assume the origin is in the lower region and the destination in the upper one. Then the optimal path has three segments;*
 1. *It goes straight horizontally from $\bar{\mathbf{x}}^O$ to M ,*
 2. *Continues as long as possible along M , i.e. until it reaches the y coordinate of the destination,*
 3. *At that point it goes straight horizontally from M to \mathbf{x}^D .*

Proof.- The proofs of (i) and of (ii) are the same. Consider an alternative route γ_C . Let $\bar{\mathbf{x}}$ be some point in $\gamma_C \cap M$. The proof now follows by applying the previous proposition to obtain first the optimal path between the origin and $\bar{\mathbf{x}}$ and second, the optimal path between $\bar{\mathbf{x}}$ and the destination. ■

Case 2: The function U is positive in the upper area and negative in the lower area.

This case turns out to be more complex than the previous one. The curve M has some obvious repelling properties which we state next, but they are not as general as the attractor properties that we had in the previous case.

⁸ By the maximum we mean the following. If γ_1 does not intersect M then $\hat{\gamma} = \gamma_1$. If it intersects M then $\hat{\gamma}$ agrees with γ_1 over the path segments where γ_1 is in the upper region and otherwise agrees with M . The minimum is defined similarly

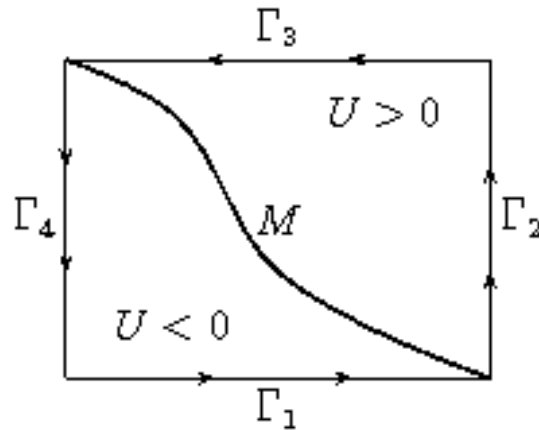


Fig. 8. Two regions separated by the curve M . Case 2.

Proposition 4. *Assume that both source and destination are in the same region. Then the paths that are optimal in Theorem 1 are optimal here as well if we restrict to paths that remain in the same region.*

Proof.- Given that the source and destination are in a region we may change the cost over the other region so that it has the same sign over all the region R . This does not influence the cost of path restricted to the region of the source-destination pair. With this transformation we are in the scenario of Theorem 1 which we can then apply. ■

Discussion.- Note that the (sub)optimal policies obtained in Proposition 4 indeed look like being repelled from M ; their two segments trajectory guarantees to go from the source to the destination as far as possible from M .

We note that unlike the attracting structure that we obtained in Case 1, one cannot extend the repelling structure to the case where the paths are allowed to traverse from one region to another.

7 User optimisation and congestion dependent cost

We go beyond the approach of geometrical optics by allowing the cost to depend on congestion. Shortest path costs can be a system objective as we shall motivate below. But it can also be the result of decentralised decision making by many “infinitesimally small” players where a player may represent a single packet (or a single session) in a context where there is a huge population of packets (or of sessions). The result of such a decentralised decision making can be expected to satisfy the following properties which define the so called, user (or Wardrop) equilibrium:

“Under equilibrium conditions traffic arranges itself in congested networks such that all used routes between OD pair (origin-destination pair), have equal and minimum costs while all unused routes have greater or equal costs” [30].

Related work.- Both the framework of global optimisation as well as the one of minimum cost path had been studied extensively in the context of road traffic engineering. The use of a continuum network approach was already introduced on 1952 by Wardrop [30] and by Beckmann [4]. For more recent papers in this area, see e.g. [6,7,14,16,32] and references therein. We formulate it below and obtain some of its properties.

Motivation.- One popular objective in some routing protocols in ad-hoc networks is to assign routes for packets in a way that each packet follows a minimal cost path (given the others’ paths choices) [11]. This has the advantage of equalising source-destination delays of packets that belong to the same class, which allows one to minimise the amount of packets that come out of sequence. (This is desirable since in data transfers, out of order packets are misinterpreted to be lost which results not only in retransmissions but also in drop of systems throughput.)

Traffic assignment that satisfies the above definition is known in the context of road traffic as Wardrop equilibrium [30].

Congestion dependent cost

We now add to c_1 the dependence on T_1 and to c_2 the dependence on T_2 , as in Section 4. Let $V^j(\mathbf{x})$ be the minimum cost to go from a point \mathbf{x} to B^j at equilibrium. Equation (12) still holds but this time with c_i that depends on T_i^j $i = 1, 2$, and on the total flows T_i $i = 1, 2$. Thus (13) becomes, $\forall j \in \{1, \dots, \nu\}$,

$$0 = \min_{i=1,2} \left(c_i(\mathbf{x}, T_i) + \frac{\partial V^j(\mathbf{x})}{\partial x_i} \right), \quad \forall \mathbf{x} \in B^j, V^j(\mathbf{x}) = 0. \quad (14)$$

We note that if $T_i^j(\mathbf{x}) > 0$ then by the definition of the equilibrium, i attains the minimum at (14). Hence (14) implies the following relations for each traffic class j , and for $i = 1, 2$:

$$c_i(\mathbf{x}, T_i) + \frac{\partial V^j}{\partial x_i} = 0 \quad \text{if } T_i^j > 0, \quad (15a)$$

$$c_i(\mathbf{x}, T_i) + \frac{\partial V^j}{\partial x_i} \geq 0 \quad \text{if } T_i^j = 0. \quad (15b)$$

This is a set of coupled PDE's, actually difficult to analyse further.

Beckmann transformation

As Beckmann et al. did in [5] for discrete networks, we transform the minimum cost problem into an equivalent global minimisation one. We shall restrict here to the single class case. To that end, we note that equations (15a)-(15b) have exactly the same form as the Kuhn-Tucker conditions (6a)-(6b), except that $c_i(\mathbf{x}, T_i)$ in the former are replaced by $\partial g(\mathbf{x}, \mathbf{T})/\partial T_i(\mathbf{x})$ in the latter. We therefore introduce a *potential function* ψ defined by

$$\psi(\mathbf{x}, \mathbf{T}) = \sum_{i=1,2} \int_0^{T_i} c_i(\mathbf{x}, s) ds$$

so that for both $i = 1, 2$:

$$c_i(\mathbf{x}, T_i) = \frac{\partial \psi(\mathbf{x}, \mathbf{T})}{\partial T_i}.$$

Then the user equilibrium flow is the one obtained from the global optimisation problem where we use $\psi(\mathbf{x}, \mathbf{T})$ as local cost. Hence, the Wardrop equilibrium is obtained as the solution of

$$\min_{T(\cdot)} \int_{\Omega} \psi(\mathbf{x}, \mathbf{T}) d\mathbf{x} \quad \text{subject to } \nabla \cdot \mathbf{T}(\mathbf{x}) = \rho(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega.$$

In the special case where costs are given as a power of the flow as defined in eq. (7), we observe that equations (15a)-(15b) coincide with equations (8a)-(8b) (up-to a multiplicative constant of the cost). We conclude that for such costs, the user equilibrium and the global optimisation solution coincide.

8 Numerical Example

The following example is an adaptation of the road traffic problem solved by Dafermos in [6] to our ad-hoc setting. We therefore use the notation of [6] for the orientation, as we did in Section 4. Thus the direction from North to South will be our positive x_1 axis, and from West to East will be the positive x_2 axis. The framework we study is the user optimisation with congestion cost. For each point on the West and/or North boundary we consider the point to boundary problem. We thus seek a Wardrop equilibrium where each user can choose its destination among a given set. A flow configuration is a Wardrop equilibrium if under this configuration, each origin chooses a destination and a path to that destination that minimise that users cost among all its possible choices.

Consider the rectangular area R on the bounded domain Ω defined by the simple closed curve $\partial R^+ = \Gamma_1^+ \cup \Gamma_2^+ \cup \Gamma_3^- \cup \Gamma_4^-$ where

$$\begin{aligned} \Gamma_1 &= \{0 \leq x_1 \leq a, \quad x_2 = 0\}, \Gamma_2 = \{x_1 = a, \quad 0 \leq x_2 \leq b\}, \\ \Gamma_3 &= \{0 \leq x_1 \leq a, \quad x_2 = b\}, \Gamma_4 = \{x_1 = 0, \quad 0 \leq x_2 \leq b\}. \end{aligned}$$

Assume throughout that $\rho = 0$ for all $\mathbf{x} \in \mathring{\Omega}$, and that the costs of the routes are linear, i.e.

$$c_1 = k_1 T_1 + h_1 \quad \text{and} \quad c_2 = k_2 T_2 + h_2, \tag{16}$$

with $k_1 > 0, k_2 > 0, h_1,$ and h_2 constant over Ω .

We are precisely in the framework of section 7 and 4 with affine costs per packet. As a matter of fact, the potential function associated with these costs is

$$\Psi(\mathbf{T}) = \sum_{i=1}^2 \int_0^{T_i} (k_i s + h_i) ds = \sum_{i=1}^2 \left(\frac{1}{2} k_i T_i + h_i\right) T_i.$$

Now, we want to handle a condensation of sources or sinks along the boundary. While this is feasible with the framework of section 4, it is rather technical. We rather use a more direct path below.

Notice that we have in $\mathring{\Omega}$, we have

$$\frac{\partial T_1}{\partial x_1} + \frac{\partial T_2}{\partial x_2} = 0.$$

Take any closed path γ surrounding a region ω . Then by Green formula,

$$\oint_{\gamma} T_1 d\xi_2 - T_2 d\xi_1 = \int_{\omega} \frac{\partial T_1}{\partial x_1} + \frac{\partial T_2}{\partial x_2} = 0$$

Therefore we can define

$$\phi(\mathbf{x}) := \int_{\mathbf{x}^0}^{\mathbf{x}} T_1 d\xi_2 - T_2 d\xi_1$$

the integral will not depend on the path between \mathbf{x}^0 and \mathbf{x} and ϕ is thus well defined, and we have

$$\frac{\partial \phi(\mathbf{x})}{\partial x_2} = T_1(\mathbf{x}) \quad \frac{\partial \phi(\mathbf{x})}{\partial x_1} = -T_2(\mathbf{x}). \tag{17}$$

We now make the assumption that there is sufficient demand and that the congestion cost is not too high so that at equilibrium the traffic T_1 and T_2 are strictly positive over all Ω [6]. It turns out that all paths to the destination are used. Thus, from Wardrop’s principle, the cost $\int \mathbf{c} d\mathbf{x}$ is equalised between any two paths. And therefore,

$$\frac{\partial c_1}{\partial x_2} = \frac{\partial c_2}{\partial x_1}.$$

Using the equations in (16) then

$$k_1 \frac{\partial T_1}{\partial x_2} = k_2 \frac{\partial T_2}{\partial x_1},$$

and from equations in (17) we have

$$k_1 \frac{\partial^2 \phi}{\partial x_2^2} + k_2 \frac{\partial^2 \phi}{\partial x_1^2} = 0.$$

Let $k_i = K_i^2$. Divide the above equation by $k_1 k_2$. One obtains

$$\frac{1}{K_1^2} \frac{\partial^2 \phi}{\partial x_1^2} + \frac{1}{K_2^2} \frac{\partial^2 \phi}{\partial x_2^2} = 0.$$

Following the classical way of analysing the Laplace equation, (see[31]) we attempt a separation of variables according to

$$\phi(x_1, x_2) = F_1(K_1 x_1) F_2(K_2 x_2).$$

We then get that

$$\frac{F_1''(K_1 x_1)}{F_1(K_1 x_1)} = -\frac{F_2''(K_2 x_2)}{F_2(K_2 x_2)} = s^2.$$

In that formula, since the first term is independent on x_2 and the second on x_1 , then both must be constant. We call s^2 that constant, but we do not know its sign. Therefore, s may be imaginary or real. All solutions of this system for a given s are of the form

$$F_1(x) = A \cos(isx) + B \sin(isx), \quad F_2 = C \cos(sx) + D \sin(sx).$$

As a matter of fact, ϕ may be the sum of an arbitrary number of such multiplicative decompositions with different s . We therefore arrive at general formula such as

$$\phi(x_1, x_2) = \int [A(s) \cos(isK_1x_1) + B(s) \sin(isK_1x_1)][C(s) \cos(sK_2x_2) + D(s) \sin(sK_2x_2)] ds.$$

From this formula, we can write T_1 and T_2 as integrals also. The flow T at the boundaries should be orthogonal to the boundary, and have the local source density for inward modulus (it is outward at a sink). There remains to expand these boundary conditions in Fourier integrals to identify the functions A , B , C , and D . (Surely not a simple matter!) (It is advisable to represent the integrals of the boundary densities as Fourier integrals, since then the boundary conditions themselves will be of the form $s \int R(s) ds$, closely matching the formula we obtain for the T_i 's.)

9 Conclusions

Routing in ad-hoc networks have received much attention in the massively dense limit. The main tools to describe the limits had been electrostatics and geometric optics. We exploited another approach for the problem that has its roots in road traffic theory, and presented both quantitative as well as qualitative results for various optimisation frameworks.

Acknowledgement

We wish to thank Dr. Stavros Toumpis for helpful discussions. The work has been kindly supported by the BIONETS European Contract.

References

1. Eitan Altman, Pierre Bernhard and Alonso Silva, "The Mathematics of Routing in Massively Dense Ad-Hoc Networks", 7th International Conference on Ad-Hoc Networks and Wireless, September 10 - 12, 2008, Sophia Antipolis, France.
2. F. Baccelli, B. Blaszczyszyn and P. Muhlethaler, "An ALOHA protocol for multihop mobile wireless networks," *IEEE Transactions on Information Theory*, Vol. 52, Issue. 2, 421- 436, February 2006
3. Martino Bardi and Italo Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhauser, 1994.
4. M. Beckmann, "A continuum model of transportation," *Econometrica* 20:643-660, 1952.
5. M. Beckmann, C. B. McGuire and C. B. Winsten, *Studies in the Economics and Transportation*, Yale Univ. Press, 1956.
6. Stella C. Dafermos, "Continuum Modeling of Transportation Networks," *Transpn Res.* Vol. 14B, pp 295-301, 1980.
7. P. Daniele and A. Maugeri, "Variational Inequalities and discrete and continuum models of network equilibrium protocols," *Mathematical and Computer Modelling* 35:689-708, 2002.
8. Wendell H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. Springer, Second Edition, 2006.
9. M. Franceschetti, O. Dousse, D. Tse, P. Thiran, "Closing the gap in the capacity of random wireless networks," In *Proc. Inf. Theory Symp. (ISIT)*, Chicago, IL, July 2004.
10. M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. on Networking*, vol 10, no 4, August 2002.
11. P. Gupta and P. R. Kumar, "A system and traffic dependent adaptive routing algorithm for ad hoc networks," *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 2375-2380, San Diego, December 1997.
12. P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. IT-46, no. 2, pp. 388-404, March 2000.
13. G. A. Gupta and S. Toumpis, "Optimal placement of nodes in large sensor networks under a general physical layer model," *IEEE Secon*, Santa Clara, CA, September 2005.
14. H.W. Ho and S.C. Wong, "A Review of the Two-Dimensional Continuum Modeling Approach to Transportation Problems," *Journal of Transportation Systems Engineering and Information Technology*, Vol.6 No.6 P.53-72, 2006.

15. E. Hyytia and J. Virtamo, "On load balancing in a dense wireless multihop network," *Proceeding of the 2nd EuroNGI conference on Next Generation Internet Design and Engineering*, Valencia, Spain, April 2006.
16. G. Idone, "Variational inequalities and applications to a continuum model of transportation network with capacity constraints," *Journal of Global Optimization* 28:45–53, 2004.
17. P. Jacquet, "Geometry of information propagation in massively dense ad hoc networks," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, New York, NY, USA, 2004, pp. 157–162, ACM Press.
18. Jerrold E. Marsden and Anthony J. Tromba, *Vector Calculus*, Third Edition, W.H. Freeman and Company 1988.
19. A. MIELE, *Extremization of Linear Integrals by Green's Theorem*, Optimization Technics, Edited by G. Leitmann, Academic Press, New York, pp. 69–98, 1962.
20. Venkatesh Ramaiyan, Anurag Kumar, Eitan Altman, "Jointly Optimal Power Control and Hop Length for a Single Cell, Dense, Ad Hoc Wireless Network," *Proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt*, April 2007.
21. V. Rodoplu and T. H. Meng, "Bits-per-Joule capacity of energy-limited wireless networks," *IEEE Transactions on Wireless Communications*, Vol. 6, Number 3, pp. 857–865, March 2007.
22. I. Stavrakakis R. Ramanathan C. Santinavez, B. McDonald, "On the scalability of ad hoc routing protocols," *IEEE INFOCOM*, vol. 3, pp. 1688–1697, 2002.
23. Alonso Silva, Eitan Altman and Pierre Bernhard, "Numerical solutions of Continuum Equilibria for Routing in Dense Ad-hoc Networks", *Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communication Systems (InterPerf)*, Athens, Greece, October, 2008.
24. L. Tassiulas and S. Toumpis, "Packetostatics: Deployment of massively dense sensor networks as an electrostatic problem," *IEEE INFOCOM*, vol. 4, pp. 2290–2301, Miami, March 2005.
25. L. Tassiulas and S. Toumpis, "Optimal deployment of large wireless sensor networks," *IEEE Transactions on Information Theory*, Vol. 52, No. 7, pp 2935–2953, July 2006.
26. S. Toumpis, "Optimal design and operation of massively dense wireless networks: or how to solve 21st century problems using 19th century mathematics," in *interperf '06: Proceedings from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications systems*, New York, NY, USA, 2006, p. 7, ACM Press.
27. S. Toumpis, "Mother nature knows best: A survey of recent results on wireless networks based on analogies with physics," Unpublished.
28. S. Toumpis, R. Catanuto and G. Morabito, "Optical routing in massively dense networks: Practical issues and dynamic programming interpretation," *Proc. of IEEE ISWCS 2006*, September 2006.
29. T. V. Truong A. Ephremides, "Scheduling broadcasts in multihop radio networks," *IEEE INFOCOM*, vol. 38, pp. 456–460, 1990.
30. J.G. Wardrop, "Some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, Part II, I:325–378, 1952.
31. H. Weinberger, *A First Course in Partial Differential Equations with Complex and Transform Methods*, Dover Books on Mathematics.
32. S.C.Wong, Y.C.Du, J.J.Sun and B.P.Y.Loo, "Sensitivity analysis for a continuum traffic equilibrium problem," *Ann Reg Sci* 40:493–514., 2006.

10 Appendix: Mathematical Tools

Theorem 3 (Green's Theorem). *Let $\Omega \subseteq X$ be a region of the space, and let Γ be its boundary. Suppose that $P, Q \in C^1(\Omega)$ (We denote $C^1(\Omega)$ the set of functions that are differentiable and whose partial derivatives are continuous on Ω .) Then*

$$\oint_{\Gamma^+} Pdx + Qdy = \int_{\Omega} \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy. \quad (18)$$

Scale-Free Networks

Petri Mähönen, Frank Oldewurtel, Janne Riihijärvi

Rheinisch-Westfälische Technische Hochschule Aachen,
Department of Wireless Networks,
D-52072 Aachen, Germany
{pma, foo, jar}@mobnets.rwth-aachen.de

Abstract. We discuss the role of scale-free network models both in relational and spatial contexts. An overview of the objectives and applications of network modelling by probabilistic means is given, followed by a concise review of the state of the art in fixed network models focusing on graph-theoretic constructs. We then report on our recent work in characterising spatial structures of wireless networks by means of node location correlations. We observe a scale-free phenomenon in an experimental data set of wireless LAN access point locations. We conclude by discussing the significance of this observation as well as issues in modelling and generation of node location distributions with internal structure similar to the observed one.

1 Introduction

During the past decade or so study of the topological structure of networks has gained considerable attention. Availability of detailed measurements of the structure of a wide variety of networks has made it possible to both analyse network topologies in detail, as well as to come up with probabilistic models capturing the main features observed in those networks. Two examples of common features found in these studies are the *small-world phenomenon* [40] and that many networks are in some sense *scale-free*. The small-world phenomenon in essence means that the typical path lengths in the network are “short” compared to the total number of nodes in the network. Here being short should not be understood in absolute terms, but as logarithmic or sub-logarithmic scaling of the path length as a function of the total number of nodes. Being scale-free on the other hand means that some aspect of the network structure displays self-similar or fractal properties. Such properties are characterised by power laws, since homogeneous polynomials are only functions that are *scale-invariant*, that is, for all constants c we have $f(c \cdot x) \propto f(x)$. Vast number of natural phenomena have been noted to exhibit scale-free characteristics, so it is not surprising that network applications exist as well. We shall highlight two of these. First, we discuss scale-free structures observed in relational networks (modelled by graphs). Here the scale-free property is typically manifest in the *degree distribution* of the nodes. The second example of scale-free behaviour we shall discuss is a type of *spatial* scale-free property of node location distributions, akin to fractals.

The rest of this chapter is now structured as follows. We shall first formulate the construction of probabilistic network models in slightly more detail. We shall then discuss some particularly popular models of scale-free relational networks, namely preferential attachment models. We shall also make some cautionary comments on the limits of applicability of such models for designed networks such as the Internet routing graph. From the relational models we shall then move on to the spatial case, outlining some of our recent research results illustrating the emergence of scale-free or fractal behaviour in wireless hot-spot networks. Techniques for constructing probabilistic spatial network models are discussed, and finally the conclusions are drawn.

2 Scale-free relational network models

Fixed networks are usually modelled using various types of random graph models, that is, ensembles $(\mathcal{G}, \mathbb{P}_{\mathcal{G}})$ of graphs in some large graph space \mathcal{G} . The “modelling” comes in via the definition of the probability measure $\mathbb{P}_{\mathcal{G}}$ of a given graph to be realised. Until recently $\mathbb{P}_{\mathcal{G}}$ was invariably chosen either based on intuition or mathematical convenience. This led to the introduction of several *random graph models* which we shall discuss below. As measurements on network characteristics directly related to

graph topologies were finally conducted these models were found to be a poor match in several respects. Examples of graph metrics which differ significantly between naive random graph models and communication networks are the clustering coefficient (conditional probability that two nodes sharing a neighbour are directly connected) and the distribution and correlations of node degrees. Improved models, such as *small worlds* [40] and *scale-free networks* were designed, trying to match *qualitatively* some of the features uncovered by measurements. Here by a qualitative match we mean that the models replicate some aspects of observations, such as the scale-free character of the degree distribution, but do not accurately replicate, for example, the exponent of the related power-law. Of these models, we shall discuss the scale-free case at some length.

The first relational random network models adopted were the random graph ones of Erdős and Rényi [19], denoted by $G(n, M(n))$ and $G(n; p(n))$. Of these, $G(n, M(n))$ is the model on the collection of all graphs of order n and size $M(n)$ with uniform probability distribution whereas $G(n; p(n))$ consists of graphs of order n with each disjoint vertex pair being an edge with probability $p(n)$. Especially the model $G(n; p(n))$ has been widely applied due to its extreme simplicity. The only parameters to decide are the total number of nodes or vertices, and the connection probability p . In both cases M and p are (possibly constant) functions of n . One of the principal characteristics of a relational network model is the *degree distribution* which for the Erdős-Rényi model is given by

$$\mathbb{P} \{ d(v) = k \mid v \in V(G_{n;p}) \} = \binom{n-1}{k} p^k (1-p)^{n-1-k}, \quad (1)$$

that is, the degrees are binomially distributed. Standard approximation then gives that the degree distribution becomes Poisson in the large n limit, provided that $p(n)$ is decreased suitably (for example, by setting $\lambda \equiv (n-1)p \equiv \text{constant}$). Thus it is intuitively clear that the maximum and minimum degrees $\Delta(G_{n;p})$ and $\delta(G_{n;p})$ are very “tightly” distributed for large n . Perhaps somewhat surprisingly this is one of major shortcomings of $G(n; p)$ as a realistic network model. The second major problem is the complete independence of the edges, which clearly cannot hold in most network or lower layer applications.

As highlighted in [6] many networks are *scale-free*, meaning that their degree distributions either have a power-law tail, or at least follow a power law over several orders of magnitude. Communication networks are not an exception to this, as scale-free behaviour has been observed both in the Internet [12,20,39,21] and in the World-Wide Web [2,7,10] and also in some application-specific connection patterns, for example, in email networks [18]. Random graphs, on the other hand, have as we have argued very tightly concentrated degrees. The original scale-free graph model was given by Barabási and Albert in [6]¹. It can be described as a graph process (G_t) featuring *growth* and *preferential attachment*. The process starts with an initial graph G_0 . Then at each step a new vertex is added with m edges to other vertices, probability of i being selected as the endpoint given by $\Pi(i) \equiv d(i)/\sum_j d(j)$. At time t , the process has produced a graph of $t + |V(G_0)|$ vertices and $mt + |E(G_0)|$ edges. A common modification of the process is to start with a graph with no edges, and make the attachment probability proportional to $d(i) + 1$ instead of $d(i)$. An example of a graph with $m = 1$ at $t = 1250$ starting with G_0 of five vertices and no edges is shown in Figure 1.

Barabási and Albert gave the following heuristic argument (called *continuum approach*) for their model yielding degree distribution of the form $p_k \sim k^{-3}$. Suppose degrees $d(i)$ are continuous instead of discrete variables. By the preferential attachment rule

$$\frac{\partial d_i(t)}{\partial t} = m\Pi(i) = \frac{d_i}{2t}. \quad (2)$$

Solving this we have $d_i(t) = m\sqrt{t/t_i}$, where t_i is the time i was created. Thus $\mathbb{P} \{ d_i(t) > k \} = \mathbb{P} \{ t_i < tm^2/k^2 \} = m^2/k^2$. Differentiating this gives $\mathbb{P} \{ d_i = k \} = 2m^2/k^3$. Much more is actually known on the structure of the graphs arising from different variations of preferential attachment models. See, for example, [28] for an extensive collection of results. Recently the “classical” graph theory community as also become interested in scale-free random graphs. For example, Bollobás et al. [9,8] have given a rigorously constructed

¹ It should be pointed out that Price gave a very similar model much earlier in [36], but this has remained practically unknown for the applied graph theory community until recently.

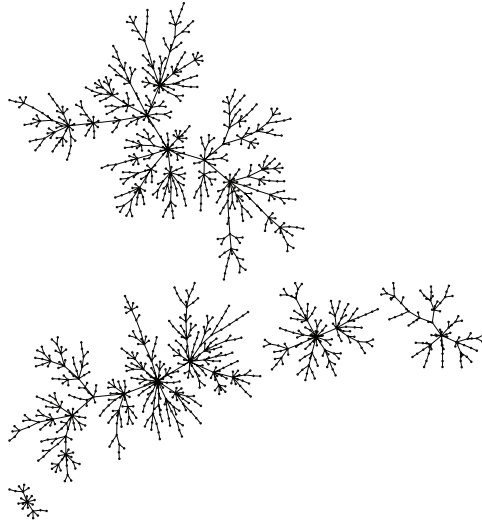


Fig. 1. An example realisation of the model of Barabási and Albert with $m = 1$.

variant of the model of Barabási and Albert, and shown that it features the properties derived above. In particular, their *LCD-model* has a degree sequence following a k^{-3} power law with path lengths of order $\ln n / \ln \ln n$. See also [11] and [14].

While growth with variants of preferential attachment is by far the most studied combination in dynamic network models, other types of dynamics have been studied as well. In [1] effects of local events (such as edge rewirings) in combination with growth were studied. Effects of constraints such as maximum node lifetimes and capacities have been studied by several authors [16,17,4]. An interesting model based on growth and *copying* has been given in [27,29] to help to explain the structure of the World-Wide Web graph. A salient feature of this model is that it can produce scale-free degree distributions for all values of the power law exponent in $[2, \infty[$. See [13] for further discussion on the copying models.

We conclude this section with a word of caution. While the models described above certainly replicate many of the degree-related properties observed in communication networks, the case of modelling the large-scale structure of the Internet is by no means closed. Several features observed related to, for example, the spectral structure and correlations in the node degrees are not properly covered by the preceding models. An interesting take on the subject has been emerging from Doyle's HOT programme (see [31,3,41]), trying to explain the structure of the present-day networks as an outcome of a large collection of local optimisation procedures. Nevertheless, their applications have also been mainly confined to toy-models, leaving ample room for further work on these matters.

3 Scale-free spatial structures

Relational structures are obviously of major importance in studying fixed networks. In wireless networks the situation is, however, slightly different as the relational structure is purely a derived quantity, dependent on the *spatial structure* of the network.

The spatial structure of the network can be modelled using the theory of *point processes*. Intuitively point processes yield random point patterns in the same sense as random variables yield random numbers (see the monographs of Karr [24] and Stoyan et al. [38] for an introduction). Technically a point process N on a region A is defined as a random counting measure, that is, it assigns a random natural number $N(A_i)$ to any suitably regular subset A_i of A . The natural interpretation we are after is to think of N as a random distribution of indistinguishable points in A , $N(A_i)$ being the number of points lying in A_i . We can always write $N = \sum_i \varepsilon_{X_i}$, where ε_x is the point mass at x (the measure equivalent to Dirac delta distribution), and X_i is an A -valued random variable. The most fundamental point process is undoubtedly the *Poisson point process* with intensity measure μ . It is defined by requiring independence of $N(A_i)$ for

all disjoint A_i and by the law

$$P\{N(A) = k\} = \frac{\mu(A)^k}{k!} \exp(-\mu(A)).$$

Conditioned on the total number of points, the Poisson point process is used almost exclusively as the model for node location distributions in the literature. It can clearly be thought of as the spatial analog of the relational model of Erdős and Rényi. Unfortunately, as was the case above, the Poisson point process cannot faithfully represent the rich structure present in the actually deployed wireless networks [37], making its universal application problematic. We shall now have a more detailed look at the issues just alluded to, with more detailed discussion available in [32].

The scale-free character of the location distributions of wireless nodes lies in spatial correlation functions. Rigorously correlations of spatial processes is usually approached using Palm distributions. Here we take a different road adopted in the applied spatial statistics community (astrophysicists in particular), defining the *pair correlation function* via the definition of the joint probability density

$$dP = \nu^2(1 + \xi(r)) dA_1 dA_2 \quad (3)$$

of finding one point in each of the two area elements dA_1 and dA_2 with ν being the intensity of N . Clearly if $\xi \equiv 0$ no obvious correlation is present, as one would expect for, for example, the Poisson point process. Positive values of ξ indicate positive correlations at the corresponding distance scales. The pair-correlation function does not, of course, determine the distribution of a point process uniquely. An illustrative example is the process of Baddeley and Silverman [5] having the same second-order structure as the Poisson point processes while being structurally substantially different. Nevertheless, it has been found extremely powerful tool in a number of sciences and by the virtue of being straightforward to estimate numerically, warrants close attention.

The definition of the pair-correlation function can, of course, be extended to the case of n -point correlations. The definition analogous to (3) is given by the joint probability density

$$dP = \nu^n(1 + \xi^{(n)}(r_{1,2}, r_{1,3}, \dots, r_{n-1,n})) dA_1 \cdots dA_n,$$

of finding a point in each of the area elements dA_i , where $r_{i,j}$ denotes the distance between area elements dA_i and dA_j . For example, considering correlations up to the three-point case yields

$$\begin{aligned} dP &= \nu^3(1 + \xi^{(3)}(r_{1,2}, r_{1,3}, r_{2,3})) dA_1 dA_2, dA_3 \\ &= \nu^3(1 + \xi(r_{1,2}) + \xi(r_{1,3}) + \xi(r_{2,3}) + \zeta(r_{1,2}, r_{1,3}, r_{2,3})) dA_1 dA_2, dA_3, \end{aligned} \quad (4)$$

where ζ is the *reduced three-point correlation function*, expressing the residual three-point correlations that do not arise from the two-point contributions directly.

To estimate the n -point correlations from a spatial data set a simple binning procedure can be utilised. First, define the *pair-counting function* [26] by

$$\Phi_r(x, y) \equiv [r \leq d(x, y) \leq r + \Delta], \quad (5)$$

where d is the metric on E , and $\Delta \in \mathbb{R}_+$ is the width of the radial bin. Let D denote the collection of observed point locations with $n \equiv |D|$, and define the *normalised observed pair counts* by

$$DD(r) = \sum_{x \in D} \sum_{y \in D} [x \neq y] \Phi_r(x, y) / (n(n-1)). \quad (6)$$

Further, let R be a realisation of a binomial point process of m points and define the *normalised cross-pair and random-pair counts* by

$$DR(r) = \sum_{x \in D} \sum_{y \in R} \Phi_r(x, y) / (nm) \quad (7)$$

and

$$RR(r) = \sum_{x \in R} \sum_{y \in R} [x \neq y] \Phi_r(x, y) / (m(m-1)), \quad (8)$$

respectively.

We can now write down the classical *pairwise estimators* for $\xi(r)$. The *natural estimator*, the *Davis and Peebles estimator* [15] and the *Hewett estimator* [23] are given by the expressions

$$\hat{\xi}_N \equiv \frac{DD}{RR} - 1, \quad \hat{\xi}_{DP} \equiv \frac{DD}{DR} - 1 \quad \text{and} \quad \hat{\xi}_{He} \equiv \frac{DD - DR}{RR}. \quad (9)$$

More recently new estimators have been suggested by *Hamilton* [22] and *Landy and Szalay* [30], which are given by

$$\hat{\xi}_{Ha} \equiv \frac{DDRR}{DR^2} - 1 \quad \text{and} \quad \hat{\xi}_{LS} \equiv \frac{DD - 2DR + RR}{RR}. \quad (10)$$

In addition to the pairwise estimators a number of *geometric estimators* have been considered in the literature. A survey of these estimators as well as an experimental evaluation of their accuracy and the accuracy of the various pairwise estimators introduced above can be found from [26]. There it is concluded that the Landy-Szalay pairwise estimator $\hat{\xi}_{LS}$ has by quite some margin the most satisfactory overall performance. The main differences in the above estimators arise in terms of estimation bias and effectiveness in terms of dealing with edge effects (that is, being able to estimate $\xi(r)$ correctly even if only a subset of the point pattern is observed). For most applications the differences are relatively small, but since the evaluation of $\hat{\xi}_{LS}$ is not significantly more demanding computationally than the other estimators, there is little reason not to use it in experimental work.

An application of the above estimators to the data set of the point distribution of approximately 10^6 measured Wireless LAN access point locations obtained from the WiGLE database is given in [37]. See Figure 3 for an illustration. The corresponding estimates of the two and (reduced) three-point correlation

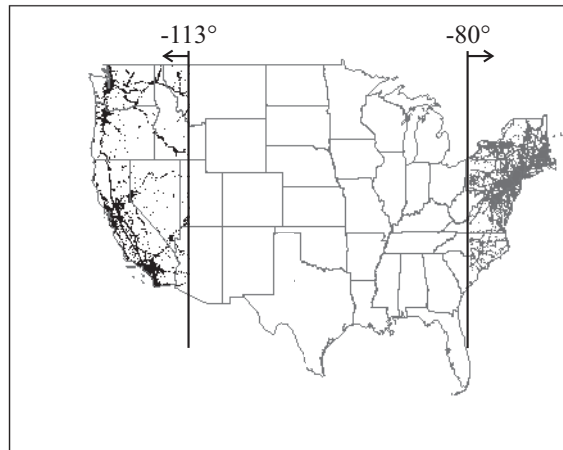


Fig. 2. The WLAN access point locations featuring scale-free spatial structure. (Adapted from [37].)

functions are shown in Figure 3. As can be seen from the figures the correlation functions follow approximately a (broken) power law. In this sense the WLAN access points can be said to form a network that is spatially scale-free in the sense of having power-law n -point correlations. In fact the power-law tendency continues to hold for higher-order correlation functions as well. The lack of such structure in commonly employed synthetic models is certainly an issue to be considered and its impact to be studied carefully. In the following section we introduce some models given in the literature that can be used to generate point distributions with scale-free characteristics in terms of correlations.

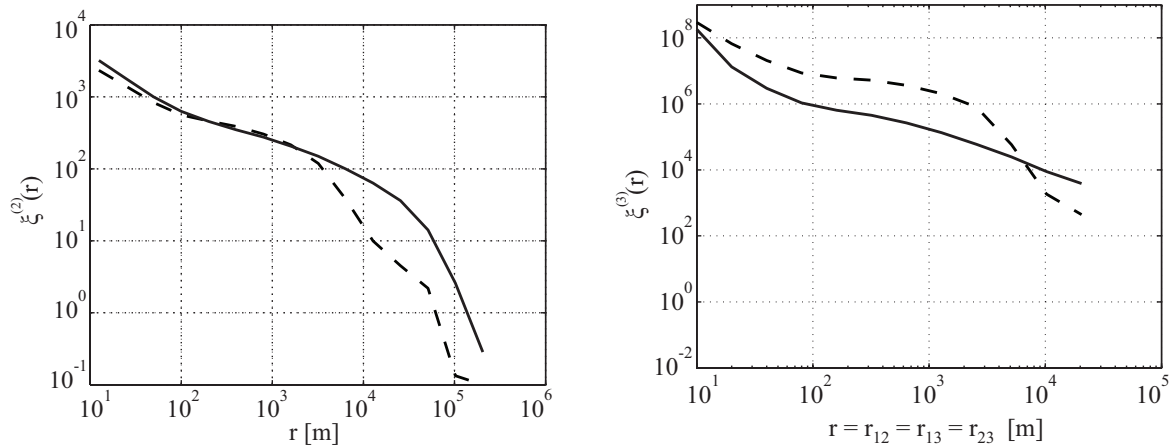


Fig. 3. The pair correlation functions of the WLAN access point locations for the east (dashed line) and west coast (solid line) data sets together with the (reduced) three-point correlation functions of the WLAN access point locations for the east (dashed line) and west coast (solid line) data sets. The triplets considered in the estimator form equilateral triangles of length r . (Adapted from [37].)

4 Modelling and generation

Some specific point processes are known which exhibit power-law n -point correlations with a fixed exponent. Example of such a process is the *segment Cox process*, see, for example, [35] for a precise definition and discussion in an applied setting. Unfortunately such models are of limited use, as the power law exponent is fixed to a predefined integer value, and more complicated correlation structures cannot be reproduced. Additionally, such models offer no insight into the possible origins of the power-law distributions arising.

The first problem is remedied by using a flexible model with the $\xi^{(n)}(r)$ as tunable parameters. Such a family of models has been given by Kerscher [25]. His models are based on *clustering*. Informally a usual clustered point process is obtained from some basic process N on E by replacing the points of N by realisations of further point processes N_i . The general and rigorous definition is obtained by the means of marked point processes. One considers a marked point process with N as the underlying process with marks N_i on the space of point processes of another well-behaved topological space E' . The *cluster process* is then defined as the superposition of the N_i . Of course, the intuitive picture given in above is only accurate in the case $E = E'$.

In Kerscher's model a particular form of clustering, Gauss-Poisson one, is used. The processes N_i are in this case independent, with realisations consisting of either a single point at the origin, or two points, with one at the origin, and distance to the other point having distribution $g(r)$ (the direction being uniformly distributed on $[0, 2\pi)$). By tuning the inter-point distribution function $g(r)$ appropriately any n -point correlation structure can be modelled, and the probability of a single-point cluster vs. a two-point cluster can be used to tune the intensity of the resulting process. See [25] for details.

Another approach worth of consideration is the application of processes with explicit interactions between the points. Such processes could be considered as a spatial analog of the Albert-Barabasi model for relational networks. The generic framework for such processes is given by the *Gibbs models* defined as follows. Let us first consider a process N of n points $\{X_i\}$, defined on a bounded set $E \subset \mathbb{R}^d$ of volume A . Suppose further that the law of N is given by a density function $f : \mathbb{R}^{nd} \rightarrow [0, \infty)$, that is,

$$\mathbb{P}\{X_1, \dots, X_n \in B\} = \int_B f(X_1, \dots, X_n) dX_1 \cdots dX_n. \quad (11)$$

Now, to obtain a tractable family of models, we assume that to each configuration of points we assign a numerical value called the *energy* or the *Hamiltonian* $H(X_1, \dots, X_n)$. This approach is motivated by

models in physics in which such a quantity is often naturally defined. To obtain a more specific form for the probability distribution we can now appeal to the maximum entropy principle, calling for maximisation of the entropy of the system under the constraints obtained from observations or the modelling assumptions made. By this assumption we now seek to find the law of N maximising the differential entropy

$$S = - \int_{E^n} f(X_1, \dots, X_n) \ln f(X_1, \dots, X_n) dX_1 \cdots dX_n \quad (12)$$

under the constraint of constant H in expectation. This yields the density

$$f(X_1, \dots, X_n) = \exp(-H(X_1, \dots, X_n)) / Z, \quad (13)$$

where the partition function Z is given by

$$Z = \int_{E^n} \exp(-H(X_1, \dots, X_n)) dX_1 \cdots dX_n. \quad (14)$$

The point process N defined in terms of the density f is called a *Gibbs process*, and is the spatial analog of the maximum entropy graph models occasionally used to study scale-free network models [34].

As such the Gibbs model is of course still extremely general. Usually one assumes a simplified form for the Hamiltonian $H(X_1, \dots, X_n)$. A common choice considered in applications has been the *pairwise interaction model* defined by

$$H(X_1, \dots, X_n) = a_0 + \sum_{i=1}^n \psi(X_i) + \sum_{1 \leq i < j \leq n} \phi(X_i, X_j) \quad (15)$$

where $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ define the single-particle and pair contributions to H . Complex interaction patterns and correlation structures can also be modelled by selecting a more complicated form for the Hamiltonian. However, the simplest approach, and thus the one that should be covered in greater detail first, would be to use the results from pair-correlation analysis of existing networks as a guide in finding realistic pairwise interaction models. In the relational scale-free case the details have been worked out by Park and Newman [33], but in the spatial case very little existing work is available, indicating the need for further research.

5 Conclusions

While the structure of scale-free networks has been under intense investigation for almost a decade now the field is still active and little signs of stagnation can be observed. Rather complete understanding of the so-called configuration and preferential attachment models with scale-free degree distributions has emerged. However, the realism and completeness of these models has recently been questioned, launching new research avenues in the graph theory front. Comparatively less work has been done in characterising networks with significant spatial aspects in their structures, such as wireless networks of different kinds. In our work we proposed the use of location correlations as a powerful paradigm. Initial results indicate that the metrics chosen are able to convincingly distinguish between realisations of common artificial network models and experimentally observed node locations. This distinction can now also be made quantitative in terms of location correlations, and this quantification can be used as a basis for new network models.

References

1. R. Albert and A.L. Barabási. Topology of Evolving Networks: Local Events and Universality. *Physical Review Letters*, 85(24):5234–5237, 2000.
2. R. Albert, H. Jeong, and A.L. Barabasi. Diameter of the World-Wide Web. *Nature(London)*, 401(6749):130–131, 1999.
3. D. Alderson, J.C. Doyle, L. Li, and W. Willinger. Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications. *Internet Math*, 2(4):431–523, 2005.
4. L.A.N. Amaral, A. Scala, M. Barthelemy, and HE Stanley. Classes of Small-World Networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, 2000.

5. AJ Baddeley and BW Silverman. A Cautionary Example on the Use of Second-Order Methods for Analyzing Point Patterns. *Biometrics*, 40(4):1089–1093, 1984.
6. A.L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
7. A.L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, 2000.
8. B. Bollobás and O. Riordan. The Diameter of a Scale-Free Random Graph. *Combinatorica*, 24(1):5–34, 2004.
9. B. Bollobas, O. Riordan, J. Spencer, and G. Tusnady. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.
10. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, 2000.
11. P.G. Buckley and D. Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282(1-3):53–68, 2004.
12. Q. Chen, H. Chang, R. Govindan, and S. Jamin. The origin of power laws in Internet topologies revisited. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2, 2002.
13. F. Chung, L. Lu, T.G. Dewey, and D.J. Galas. Duplication Models for Biological Networks. *Journal of Computational Biology*, 10(5):677–687, 2003.
14. C. Cooper and A. Frieze. A general model of web graphs. *Random Structures and Algorithms*, 22(3):311–335, 2003.
15. M. Davis and PJE Peebles. A survey of galaxy redshifts. V- The two-point position and velocity correlations. *Astrophysical Journal*, 267:465–482, 1983.
16. SN Dorogovtsev and JFF Mendes. Evolution of networks with aging of sites. *Physical Review E*, 62(2):1842–1845, 2000.
17. SN Dorogovtsev and JFF Mendes. Scaling behaviour of developing and decaying networks. *Europhysics Letters*, 52(1):33–39, 2000.
18. H. Ebel, L.I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Physical Review E*, 66(3):35103, 2002.
19. P. Erdős and A. Rényi. On the evolution of random graphs. *Bulletin of the Institute of International Statistics*, 38:343–347, 1961.
20. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *Comput. Commun. Rev.*, 29:251–263, 1999.
21. R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3, 2000.
22. AJS Hamilton. Toward better ways to measure the galaxy correlation function. *Astrophysical Journal*, 417(1):19–35, 1993.
23. PC Hewett. The estimation of galaxy angular correlation functions. *Royal Astronomical Society, Monthly Notices*, 201:867–883, 1982.
24. A. F. Karr. *Point Processes and Their Statistical Inference*. Marcel Dekker, 2nd edition, 1991.
25. M. Kerscher. Constructing, characterizing, and simulating Gaussian and higher-order point distributions. *Physical Review E*, 64(5):56109, 2001.
26. M. Kerscher, I. Szapudi, and A.S. Szalay. A Comparison of Estimators for the Two-Point Correlation Function. *The Astrophysical Journal*, 535(1):L13–L16, 2000.
27. J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. *Proceedings of the International Conference on Combinatorics and Computing*, 6(1):6–1, 1999.
28. PL Krapivsky and S. Redner. Organization of growing random networks. *Physical Review E*, 63(6):66123, 2001.
29. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 57–65, 2000.
30. S.D. Landy and A.S. Szalay. Bias and variance of angular correlation functions. *The Astrophysical Journal*, 412:64, 1993.
31. L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the internet's router-level topology. *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–14, 2004.
32. P. Mähönen, M. Petrova, and J. Riihijärvi. Applications of topology information for cognitive radios and networks. In *Proceedings of IEEE DySPAN 2007*, April 2007.
33. J. Park and MEJ Newman. Origin of degree correlations in the Internet and other networks. *Physical Review E*, 68(2):26112, 2003.
34. J. Park and MEJ Newman. Statistical mechanics of networks. *Physical Review E*, 70(6):66117, 2004.
35. M.J. Pons-Borderia, V.J. Martinez, D. Stoyan, H. Stoyan, and E. Saar. Comparing Estimators of the Galaxy Correlation Function. *The Astrophysical Journal*, 523(2):480–491, 1999.
36. D.J.S. Price. A general theory of bibliometric and other cumulative advantage processes. *J. Amer. Soc. Inform. Sci.*, 27(5):292–306, 1976.
37. J. Riihijärvi, P. Mähönen, and M. RübSamen. Characterizing wireless networks by spatial correlations. *Communications Letters, IEEE*, 11(1):37–39, 2007.
38. D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic geometry and its applications*. Wiley, 1995.
39. A. Vázquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of the Internet. *Physical Review E*, 65(6):66130, 2002.
40. DJ Watts and SH Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–10, 1998.
41. W. Willinger, D. Alderson, J.C. Doyle, and L. Li. More "normal" than normal: scaling distributions and complex systems. *Proceedings of the 36th conference on Winter simulation*, pages 130–141, 2004.

Part III

Paradigms from Social Science

Network Formation Games

Giovanni Neglia

Institut National de Recherche en Informatique et Automatique,
INRIA Sophia Antipolis,
F-06902 Sophia Antipolis, France
giovanni.neglia@sophia.inria.fr

Abstract. Recent research in different fields like physics, economics, and social science has pointed out the important role of network structure for the performance of a distributed system (be it a group of friends, the World Wide Web or a business and commerce system). In the current Internet the network structure arises from interactions of agents at different levels. Internet Service Providers (ISPs) and different organisations decide autonomously which other entities in the Internet they want to be directly connected to. More recently, Peer-to-Peer (P2P) networks and ad hoc networks have introduced new actors shaping the current Internet structure. All these agents (ISPs, peers,...) may have disjoint and competing interests, so game theory can provide useful mathematical tools to study the outcomes of their interactions. In particular, there is a growing research trend on so-called network formation games, which explicitly consider players who can decide to connect to each other. In these games the network structure both influences the result of the economic interactions and is shaped by the decisions of the players. The purpose of this chapter is to provide the reader unfamiliar with this research area with the basic concepts, pointers to other surveys, and an overview of current results in the computer networks field.

1 Introduction

Recent research in different fields like physics, economics, social sciences has pointed out the important role of network structure for the performance of a distributed system (would it be a group of friends, the World Wide Web or a business and commerce system). The birth of a new *science of networks* has been advocated [6].

In current Internet the network structure arises from interactions of agents at different levels. First regarding the physical infrastructure of traditional wired Internet, there is no single central authority designing the graph, but Internet Service Providers (ISPs) and different organisations decide autonomously which other entities in the Internet they want to be directly connected to, and the use of the link established (e.g. if these links can forward also third party traffic and if so at which price).

New scenarios are extending the number of actors involved in shaping the current Internet structure. Peer-to-Peer (P2P) networks and ad hoc networks clearly show this trend. P2P applications rely on overlays, i.e. logical networks, built atop of the existing physical infrastructure. Connections among nodes are logical or virtual links at the application level, so an end user interacts with other users in order to create the logical network. Current overlays exhibit different topologies, like stars, rings, d-dimensional grids, butterflies, de Bruijn graphs, trees, and random graph generated topologies. The wide spectrum of proposed solutions reveals by itself that there is no optimal choice for all possible design goals like robustness, maintenance costs, efficiency for routing and efficiency for dissemination. Ad hoc networks are formed by nodes (like laptops or sensors) with wireless transmission capabilities located in a give area, in general without any backbone infrastructure to route packets. For this reason nodes have to act as routers and to cooperate to deliver traffic in the network. They can create a specific topology, because they have the possibility -to a given extent- to determine the set of reachable nodes, by deciding for example their power transmission level and the orientation of the antenna and which nodes they want to forward the traffic to among those reachable.

Endogenous network formation is probably going to play a more and more important role in future computer networks. This is a particular aspect of current attrition of central and explicit control. In fact computer networks are rapidly increasing in size and extending to new environments (vehicles, home appliances, natural ecosystems...). Central and explicit control is not a feasible solution for the management and operation of these large scale distributed systems. Even traditional approaches to distributed

system design are not applicable as they would not scale to such sizes, nor would they be able to deal with the fast dynamics and the intrinsic unreliability of the components of these systems. For example in P2P networks and ad hoc and sensor networks new nodes can join the system while others leave abruptly because of users decisions or failures (e.g. due to battery depletion). At the same time mobility and vagaries of wireless channel can frequently sever and create links. In some cases even connectivity across the network cannot be assumed: in Delay Tolerant Networks disconnected operation is the normal situation rather than an exception. These issues call for completely distributed self-organising systems, i.e. systems able to reorganise themselves on the basis of local knowledge to deal with such dynamic environment. The absence of a central server (or of central servers) intrinsically offers more robustness to failures. Also, completely decentralised systems appear intrinsically appealing to implement communication, storage and computational services in a bottom-up fashion, at a very low cost. Besides these technological motivations, the popularity of P2P applications among Internet users also strengthen this tendency to decentralisation.

In general, altruistic cooperation cannot be assumed in computer networks, where users likely have disjoint and competing interests, and this is also true for network formation. This is clear for ISPs, that are primarily concerned with maximising profits. But also users in a P2P networks can have specific interests in controlling the number of connections they establish (because they can be costly), or in establishing some connections rather than others, e.g. to increase their downloading rate in a file-sharing network, or to achieve a better quality in a multi-cast overlay for real-time contents. Also the behaviour of nodes in an ad hoc network can be driven by the wish to spare their own battery, avoiding to cooperate in the forwarding process. Decentralisation makes these issues more significant, because in a decentralised environment it is harder to force users to cooperate. Game theory is clearly a candidate to study such kind of problems, because it provides mathematical tools to study the outcomes of interactions among self-interested players . In particular there is a growing research trend in game theory on so-called network formation games, which explicitly considers players who can decide to connect to each other. In these games the network structure both influences the result of the economic interactions and is shaped by the decisions of the players. Characterising the structure and the efficiency of networks arising as equilibria from players interaction is one of the main purpose of such research.

2 Purpose and outline

The purpose of this document is not to be an exhaustive survey on network formation games techniques and applications, but rather to provide the reader unfamiliar with this research area with the basic concepts, pointers to other surveys, and an overview of current results in the computer networks field.

The chapter is organised as follows. In section 3 we introduce the two main approaches in current research on network formation games research adopted from two different different communities. In section 4 we illustrate the main concepts and tools used in this research field. Finally in section 5 we present papers addressing network formation problems in computer networks.

3 Approaches

The birth of network formation games can be dated to 1977 with Myerson's paper [24]. Since that time network formation games have been applied as models of social and economic networks focusing on pairwise relations between individuals or companies who can locally form direct links. In this research stream link creation usually requires agreement among the players, so new equilibria concepts have been developed to address this coordination requirement (see section 4). An optimal survey of this socio-economic literature is [16].

More recently there has been an increasing interest on network formation games from researchers in theoretical computer science, interested in the fusion of algorithmic ideas with concepts and techniques from game theory. This new research area is sometimes referred to as *algorithmic game theory*. Algorithmic game theory focuses mainly on computational issues in game theory, like Nash equilibrium or best response computation or mechanism design. Useful introductions to this field are [27,15], while

the first book on this subject has appeared recently [1]. Up to now algorithmic game theory has given a specific contribution to network formation games with the elaboration of price of anarchy and price of stability concepts to quantify the cost of using decentralising solution to a problem (see section 4). In [29] the author presents some illustrative results on network formation games fitting them in more general frameworks (congestion games, potential games and utility games) and introducing open research issues.

4 Main Concepts in Network Formation Games

In this section we are going to introduce some important concepts in network formation games, using a simple example as starting point.

Let us assume that there are N players and the strategy of each player is the set of links he would like to create with other players. Formally we indicate the strategy of player i with the vector \mathbf{s}_i , where $s_{i,j} = 1$ indicates his willingness to form a link with player j , $s_{i,j} = 0$ otherwise. The set of possible strategies is than $S_i = \{0, 1\}^N$. We denote the link between i and j as ij and we assume that the link between i and j is created when one of the two players wants it, i.e. if $s_{i,j} \vee s_{j,i} = 1$. Given a strategy profile \mathbf{s} , we indicate the network rising from player interaction as $g(\mathbf{s})$, which is simply the list of unordered pair of connected players, and we indicate as G the set of all possible networks users can form. With some abuse of notation, we let $ij \in g$ indicate that link ij belongs to the network g and we let also $g + ij$ denote the network found by adding the link ij to the network g and $g - ij$ denote the network found by deleting the link ij from the network g . The utility of player i is a function of the final network g , we denote it as $u_i(g)$.

For example in [13] user utility takes into account the cost of the number of connections established with other player $-|\mathbf{s}_i|$ for user i - as well as the sum of the costs of reaching all the other players:

$$-u_i(g(\mathbf{s})) = \alpha|\mathbf{s}_i| + \sum_{j=1}^N d_{(i,j)}(g(\mathbf{s})), \quad (1)$$

where $d_{(i,j)}(g(\mathbf{s}))$ is the shortest-path distance (in terms of hops-count) between nodes i and j in the graph $g(\mathbf{s})$.

4.1 Nash Equilibrium

A standard equilibrium concept in Game Theory is Nash equilibrium. With reference to our example above we say that a strategy profile \mathbf{s}^* is a Nash Equilibrium if

$$\forall i, \forall \mathbf{s}_i \in S_i, \quad u_i(g(\mathbf{s}^*)) \geq u_i(g(\mathbf{s}_i, \mathbf{s}_{-i}^*)),$$

where $(\mathbf{s}_i, \mathbf{s}_{-i}^*)$ indicates a strategy profile \mathbf{s}' such that $\mathbf{s}'_j = \mathbf{s}_j^* \forall j \neq i$, and $\mathbf{s}'_i = \mathbf{s}_i$. This is a Nash equilibrium in pure strategies, mixed strategies are not usually considered in network formation games literature.

A network g is said to be *Nash stable* if it arises from a Nash equilibrium.

Sometimes also *approximate Nash equilibria* are considered. Roughly speaking an approximate Nash equilibrium is a strategy profile, such that users convenience to defect is smaller than a given bound. In case of absolute improvement bounds, \mathbf{s}^* is a ε -approximate Nash Equilibrium (with $\varepsilon > 0$) if

$$\forall i, \forall \mathbf{s}_i \in S_i, \quad u_i(g(\mathbf{s}^*)) \geq u_i(g(\mathbf{s}_i, \mathbf{s}_{-i}^*)) - \varepsilon,$$

while in case of relative improvement bounds, \mathbf{s}^* is a ε -approximate Nash Equilibrium (with $1 > \varepsilon > 0$) if

$$\forall i, \forall \mathbf{s}_i \in S_i, \quad u_i(g(\mathbf{s}^*)) \geq \varepsilon u_i(g(\mathbf{s}_i, \mathbf{s}_{-i}^*)).$$

Approximate Nash equilibria have been advocated as tools to quantify the lack of coordination due to selfish behaviour (section 4.4) when Nash equilibria do not exist, or are hard to find.

4.2 Other Equilibria for Coordination requirement

In socio-economic literature it is common to consider situations when both players have to agree in order to create a link. For example with reference to our reference game, a link would be created only if $s_{i,j} \wedge s_{j,i} = 1$.

In such cases the concept of Nash equilibrium can be inadequate, because it allows for too many equilibria. For example the empty network is always a Nash stable network regardless of the utility functions $u_i(\cdot)$. In order to deal with this coordination requirement new equilibria concepts have been introduced. Here we mainly follow the terminology in [7], where also relations among these different equilibria are illustrated.

Pairwise Stable Networks A network g is *pairwise stable* if

- a) $\forall ij \in g, u_i(g) \geq u_i(g - ij)$,
- b) $\forall ij \notin g, \text{ if } u_i(g + ij) > u_i(g) \text{ then } u_j(g + ij) < u_j(g)$.

Then the network is not pairwise stable if some player can gain by deleting a link or two players can gain from adding a link. Note that the concept of pairwise stability does not consider more complex deviations where for example a node can sever two or more links at the same time.

Pairwise Nash Stable Networks Nash equilibrium and pairwise stability can be merged together. We say that a strategy set \mathbf{s}^* is a *pairwise Nash equilibrium* if

- a) \mathbf{s}^* is a Nash equilibrium,
- b) $\forall ij \notin g, \text{ if } u_i(g + ij) > u_i(g) \text{ then } u_j(g + ij) < u_j(g)$.

A network g is *pairwise Nash stable* if there exists a pairwise Nash equilibrium \mathbf{s} of the game, such that $g = g(\mathbf{s})$ ¹.

Allowing transfers If transfer among players are allowed than the previous definition can be straightforwardly extended. Here we only present the concept of a pairwise stable network with transfers, the reader can refer to [7] for the other extensions.

A network g is said to be *pairwise stable with transfers* if

- a) $\forall ij \in g, u_i(g) + u_j(g) \geq u_i(g - ij) + u_j(g - ij)$
- b) $\forall ij \notin g, u_i(g) + u_j(g) \geq u_i(g + ij) + u_j(g + ij)$.

4.3 Value Function and Allocation Rule

Value function and allocation rule are natural extensions of characteristic function and imputation rule from the cooperative game theory (see for example [28] for a gentle introduction to these concepts).

The value function assigns a value to every possible network players can create, $v : G \rightarrow \mathbb{R}$. The set of all possible value functions is denoted by V . The value of a network can in general depend in arbitrary ways on the structure of the networks, but component additivity and anonymity are common assumptions. A value function is *component additive* if the value of each network g is equal to the sum of the values of all the disconnected sub-graphs (components) in g ; it is *anonymous* if the value depends only on the structure and not on which player occupies a given place in this structure, i.e. on player labels ([16] for formal definitions). In the special case where the value depends only on the groups of players that are connected, the value function reduces to the characteristic function in a cooperative game.

The allocation rule is a function which specifies how the value of the network is distributed among the players, $Y : G \times V \rightarrow \mathbf{R}^N$, such that $\sum_i Y_i(g, v) = v(g)$ for all v and g . An allocation rule is *component*

¹ Sometimes definitions are not coherent across literature. For example "pairwise Nash stable networks", are referred as "pairwise equilibrium networks" in [14] and as "pairwise Nash equilibrium networks" in [11].

balanced if the value of each component is divided among players belonging to that component; it is *anonymous* if allocations do not depend on player labels.

If a network formation game is defined explicitly, defining for each player his set of strategies and his payoffs, than the value function and the allocation rule are immediately determined. For example in our simple example the value function can be defined as the sum of each node utility:

$$\begin{aligned} v(g) &= \sum_{i=1}^N u_i(g(\mathbf{s})) \\ &= - \sum_{i=1}^N \left(\alpha |\mathbf{s}_i| + \sum_{j=1}^N d_{(i,j)}(g(\mathbf{s})) \right), \end{aligned}$$

and $Y_i(g, v) = u_i(g)$ if no utility transfer is possible among players. This value function is clearly component additive and anonymous.

4.4 Price of anarchy/stability

In general the interaction of selfish peers leads to a degradation in network performance. In order to quantify such degradation it is possible to compare the value of an equilibrium network, g_{eq} with the value of an optimal network g_{opt} , i.e. a network for which $v(g_{opt})$ is maximum².

The *price of anarchy*, first defined in [17], is the ratio of the value function for the worst Nash equilibrium and that of an optimal solution. It represents a bound on the inefficiency of every possible stable outcome of the game.

The *price of stability*, first defined in [4], is the ratio of the value function for the best Nash equilibrium and that of an optimal solution. The interpretation of the price of stability is less immediate. Let us assume that there is a central authority that cannot enforce strict policies from selfish users after the network is built, but it can affect players interaction in the early stage (e.g. by introducing some form of incentives) in such a way that the final equilibrium will be the best possible Nash equilibrium. The price of stability is the minimum loss of efficiency such central authority should pay, in order to have an operation that is robust (stable) to selfish behaviours.

5 Applications to Computer Networks

In this section we present some network formation games addressing specific issues in computer networks.

We first consider games where players aim to achieve connectivity in the network. Following [29] we distinguish *local connection games* (Sec. 5.1) and *global connection games* (Sec. 5.2). In the first case each player is associated to a node and he can only decide to create links between the given node and other nodes in the network. This scenario fits the case of fully distributed P2P networks, where each peer can only decide about its local connections. In the second case a player is not associated with an individual node, but he is willing to pay for creating links in the network in order to connect some specific nodes. Finally section 5.3 introduces a few games for some specific overlays for file sharing and multi-cast.

5.1 Local Connection Games

To the best of our knowledge, the first paper studying the Internet design as a network formation game was [13]. In this paper the authors propose a simple model, where each player is a node and can create bidirectional connections to other nodes. The price of a connection is paid only by the initiator while everyone can take advantage of it. The utility function of each player is that indicated in Eq. 1. Being

² Two remarks. First, here we do not need to restrict ourselves to a specific equilibrium, g_{eq} could be a Nash stable network or one of the other equilibria introduced in section 4.2. Second, with a finite number of players the set of possible networks, G , is finite, so $v(\cdot)$ has always a maximum value on G .

that link creation does not require coordination, the authors adopt Nash equilibrium as investigation tool. They are able to characterise socially optimal networks for all the possible value of α and to determine bound for the price of stability (they do not explicitly mention it, but see results in Section 2 and the description of their game in [29]) and for the price of anarchy. They also state a tree conjecture: it exists a constant A such that for $\alpha > A$ all non-transient Nash equilibria³ are trees.

The bounds for the price of anarchy are improved in [3]. This paper proves also that the original tree-conjecture is false, but they show that for $\alpha \geq 12n \lceil \log n \rceil$ every Nash equilibrium is a tree. Moreover they extend some of their results to the case where a non-uniform traffic matrix is taken into account in players payoffs. In particular, given $w_{i,j}$ the traffic from i to j the cost of player i is:

$$-u_i(g(\mathbf{s})) = \alpha |\mathbf{s}_i| + \sum_{j=1}^N w_{i,j} d_{(i,j)}(g(\mathbf{s})).$$

Different extensions of the model in [13] are studied through simulations in [9]. In particular they introduce a node-dependent connection cost, drawn from an exponential distribution or dependant from the node degree (i.e. from his strategy), constraints on the maximum number of connections a player can open, and a underlying Internet-like topology which allows to consider as distance between nodes i and j ($d_{(i,j)}(g(\mathbf{s}))$) the latency of the path between the two nodes. The authors investigate what overlay structures arise in terms of node-degree distribution, social cost, path length, number of messages needed to build a given topology, failure and attack tolerance.

A more recent simulation study [19] considers a variant where each user can establish a fixed number of directed links and there is a non-uniform traffic matrix. The cost function is then:

$$-u_i(g(\mathbf{s})) = \sum_{j=1}^N w_{i,j} d_{(i,j)}(g(\mathbf{s})).$$

Realistic underlay topologies, deriving from topology simulators or measurements are considered. The results suggest that selfish users adopting a best response strategy are able to achieve almost optimal performance. The paper investigates also the interaction among such users and users employing more naive strategies, like connecting to a random set of players or to the closest ones. Another paper [18] from the first author of [19] addresses more theoretical issues like the existence of Nash equilibrium under uniform and non-uniform traffic matrix, the properties of Nash stable networks (differences in utilities across nodes, diameter of the graph), the possibility to reach a stable network through a best-response dynamics.

In [11] the original model is extended, considering that both players need to agree in order to create a link. Upper and lower bounds for the price of anarchy are determined for pairwise Nash stable networks⁴ and it is proven that the price of anarchy is bigger for this coordinated game than for the uncoordinated game considered in [13].

A variation of the cost function in Eq. 1 is considered in [22], where players can establish *directed* virtual links, building an overlay g on top of an existing underlay f , trying to minimise:

$$-u_i(g(\mathbf{s})) = \alpha |\mathbf{s}_i| + \sum_{j=1}^N \frac{d_{(i,j)}(g(\mathbf{s}))}{d_{(i,j)}(f)},$$

where $d_{(i,j)}(f)$ is the direct distance between node i and j in the underlay graph and $d_{(i,j)}(g(\mathbf{s}))$ is, as above the distance in the overlay graph created by players interaction. In the P2P language the ratio of these two quantities is called the *stretch* between player i and player j . While it is not clear how meaningful is minimising the sum of the stretches, the authors can prove an upper bound for the price of anarchy

³ They denote as non-transient Nash equilibria a weak Nash equilibria (i.e. one in which at least one player can change his strategy without affecting his payoff), from which there exists a sequence of single-player deviation that do not alter the player payoff, but lead to a non-equilibrium position.

⁴ Interestingly the authors proof that for their game three different equilibria concepts are equivalent: pairwise stable networks, pairwise Nash stable networks and proper equilibrium [25].

independent from the specific metric space where players can be located⁵. The paper also shows that Nash equilibrium do not always exist, but players can be trapped in an infinite loop of strategy changes (see [16] for a discussion about the existence of cycles and pairwise stable networks and sufficient conditions to exclude cycles). Finally determining the existence of a pure Nash equilibrium for this game is NP-complete.

In [12] a local connection game is considered for an ad-hoc networks scenario, where nodes can choose their power levels in order to ensure the desired connectivity properties. A main difference in comparison to the basic model in [13] is that the set of neighbours is uniquely determined by the transmission range: all the nodes inside the transmission range⁶ r of a node are its neighbours. Also in this case, once a link is established, it can be used by all other nodes, this is in our opinion the main weakness of this model, because, even if selfish and power concerned, nodes forward other nodes traffic. It is assumed that the relation between the transmitting power at node i (P_i^{emit}) and the received power at node j ($P_{i,j}^{rec}$) is $P_{i,j}^{rec} = K/d(i,j)^\gamma P_i^{emit}$, and that the received power needs to exceed a minimum level in order to have a successful transmission. In this case power minimisation from user i is equivalent to minimise r^γ . The authors consider different variants of the game: users can only try to connect to a given destination (*connectivity game* in the paper), or to all nodes (*strong connectivity game*), and they can require a single path to the destination or k node-disjoint path (*(strong) k-connectivity game*), or they can try to maximise the difference between the number of reachable nodes and r^γ . (*reachability game*). The case of directional antennas is also considered. For each of these variants the existence of a Nash equilibrium or of an approximate one is investigated. When a Nash equilibrium exists, the price of anarchy is evaluated.

5.2 Global Connection Games

In global connection games, each player can build edges throughout the network. Multiple players may share the cost of building mutually beneficial links.

The original model was proposed in [5]. The game occurs in a directed graph $G = (V, E)$, where each edge $e \in E$ has a non-negative cost c_e . The purpose of each player i ($i = 1, \dots, N$) is to connect a set of terminals to a source. A strategy of a player is a payment function p_i , where $p_i(e)$ is how much player i is willing to pay for link e . Any edge such that $\sum_i p_i(e) \geq c(e)$ is considered bought. Each player tries to minimise its total payment. The authors show that there instances of the game without deterministic Nash equilibria, but the specific case (called single source game) where each player has a single terminal and all the players share the same source always admit a Nash equilibrium. The paper evaluates the price of anarchy and the price of stability for the cases when Nash equilibria exist. Moreover it uses approximate Nash equilibria to evaluate how unhappy would the agents be if they were forced to pay for the socially optimal network. More specifically the authors try to identify approximate Nash equilibria whose total cost is within a given factor to the optimal network.

The authors of [4], together with others, consider in [5] a variant of the previous game with a specific link cost sharing among users. In the new game each user aims to connect a specific source-sink pair (s_i, t_i) , creating a path P_i in the network. The cost of each link in the path is shared equally by all the players who are interested into using that link. The final network has a social cost equal to $\sum_{e \in \{\cup_i P_i\}} c_e$. This way to share costs among the players has many interesting properties: it distributes the whole social cost (so it is *budget balanced*), it can be derived by the Shapley value and for this reason it can be shown to be the unique cost-sharing scheme satisfying a number of natural sets of axioms [23]. In this more regulated setting, it is possible to show that there always exists a Nash equilibrium of total cost at most $H(N) = \sum_{i=1}^N 1/i$ and that the price of stability is equal to $H(N)$. This result is achieved using a potential function method [21] and can be extended to some more general settings: users selecting arbitrary subsets of E (not necessarily paths), edge costs that are non-decreasing concave functions of the number of users and can include a latency cost, constraints on the number of users at each node. The authors also investigate the speed of convergence of the best-response dynamics.

⁵ Note that assuming players belong to a metric space is not trivial. In general peers in the Internet are not located in a metric space for many interesting distance metric, like for example delay (the triangle inequality is not in satisfied in general).

⁶ The transmission range is the maximum distance at which a given node can transmit successfully.

The authors of [8] extend the model in [4], by attributing a weight w_i to each player, so that the i -th player share of edge e cost is equal to $c_e w_i / \sum_j w_j$. In this case there are instances with no pure-strategy Nash equilibrium. For this reason, similarly to [5] the authors look for α -approximate Nash equilibria, whose cost is within a β factor from the optimal social cost and in particular they try to investigate how much stability one has to give up (higher α values) in order to achieve low cost solution (low β values), and vice versa. They identify a possible trade-off between β and α and show that it is very close to the best possible one. The paper presents also an interesting discussion about alternative approaches to α -approximate Nash equilibria.

5.3 Overlay Specific Games

In [30] and in its extension [26] we have proposed a network formation game to model interaction among peers using BitTorrent [10] protocol for file sharing. One of the reason of BitTorrent success is its ability to enforce cooperation among the peers (contrasting hence the well know problem of free-ride), through the so called Tit-for-Tat strategy: when a peer receives requests for file pieces from different peers, it uploads to the n_u peers (the default value is 4) from which it can download at the highest rate, i.e., its best uploaders. This strategy is clearly intended to benefit the peers who contribute more to the system. Tit-for-Tat is generally considered robust to selfish behaviour. In order to investigate this issue we have considered a game where peers can change the number of connections to open in order to improve their performance and achieve better performance. This model captures Tit-for-Tat reciprocation feature by considering that two peers set up a connection between themselves only when they both find it beneficial. For this game we have characterized the topologies of some pairwise Nash stable networks peers can form both in homogeneous scenarios and in heterogeneous scenarios (i.e. respectively when all the links have the same or different capacity values) and we have shown that loss of efficiency peers experience because of their lack of coordination is in general unbounded despite the utilisation of the Tit-for-Tat strategy. Finally we have considered a simple dynamics for this game, and have proved that when connection costs are linear functions of the number of links, this dynamics converges to a pairwise stable network. We have also quantified by simulations the convergence time and shown that as the network size increases the dynamics leads to networks near to the equilibria described.

In P2P application for multi-cast, peers are organised into an overlay content distribution tree. Each peer receives the content from his parent in the tree and distribute it to his children. Nodes with more children have clearly a higher replication burden, while nodes nearer to the source perceive in general a better service, because they experience smaller loss probability and jitter. It is hence clear that selfish peers can try to be positioned closer to the data source and to limit the number of children. In [20] the authors analyse different multi-cast protocol families and show for each of them how a peer can cheat in order to change its position and what is the impact on the global performance. In [2] a repeated-game models is proposed: each user tries to improve its position, but at the same time it wants the overlay to survive, so its utility function is the discounted future benefit over the expected lifetime of the system. Taking into account the future introduces a motivation for users to cooperate. The game is studied through simulations and some guidelines on how to make protocols more robust are proposed.

References

1. *Algorithmic Game Theory*. Cambridge University Press, 2007.
2. Mike Afegan and Rahul Sami. Repeated-game modeling of multicast overlays. In *INFOCOM*, April 2006.
3. Susanne Albers, Stefan Eilts, Eyal Even-Dar, Yishay Mansour, and Liam Roditty. On nash equilibria for a network creation game. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 89–98, New York, NY, USA, 2006. ACM Press.
4. Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 295–304, Washington, DC, USA, 2004. IEEE Computer Society.
5. Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 511–520, New York, NY, USA, 2003. ACM Press.
6. Albert-Laszlo Barabasi. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.

7. Francis Bloch and Matthew Jackson. Definitions of equilibrium in network formation games. *International Journal of Game Theory*, 34(3):305–318, October 2006. available at <http://ideas.repec.org/a/spr/jogath/v34y2006i3p305-318.html>.
8. Ho-Lin Chen and Tim Roughgarden. Network design with weighted players. In *SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 29–38, New York, NY, USA, 2006. ACM Press.
9. B. Chun, R. Fonseca, I. Stoica, and J. Kubiawicz. Characterizing selfishly constructed overlay networks. In *Proc. of IEEE INFOCOM'04, Hong Kong*, 2004.
10. Bram Cohen. Bittorrent, <http://www.bittorrent.com>.
11. J. Corbo and D. C. Parkes. The price of selfish behavior in bilateral network formation. In *Proc. of the 24th ACM Symp. on Principles of Distributed Computing*, pages 99–107, Las Vegas, Nevada, 2005.
12. Stephan Eidenbenz, V. S. Anil Kumar, and Sibylle Züst. Equilibria in topology control games for ad hoc networks. *Mob. Netw. Appl.*, 11(2):143–159, 2006.
13. A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *Proc. of the 22nd annual symposium on Principles of distributed computing*, pages 347–351, New York, NY, USA, 2003. ACM Press.
14. Sanjeev Goyal and Sumit Joshi. Unequal connections. *International Journal of Game Theory*, 34(3):319–349, October 2006. available at <http://ideas.repec.org/a/spr/jogath/v34y2006i3p319-349.html>.
15. Joseph Y. Halpern. *Computer Science and Game Theory: A Brief Survey*, 2007.
16. M. Jackson. A Survey of Models of Network Formation: Stability and Efficiency. In Gabrielle Demange and Myrna Wooders, editors, *Group Formation in Economics: Networks, Clubs, and Coalitions*. Cambridge University Press, 2004.
17. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, Trier, Germany, 4–6 March 1999.
18. Nikolaos Laoutaris, Rajmohan Rajaraman, Ravi Sundaram, and Shang hua Teng. A bounded-degree network formation game, 2007.
19. Nikolaos Laoutaris, Georgios Smaragdakis, Azer Bestavros, and John W. Byers. Implications of Selfish Neighbor Selection in Overlay Networks. In *Proceedings of IEEE INFOCOM 2007*, Anchorage, AK, May 2007.
20. Laurent Mathy, Nick Blundell, Vincent Roca, and Ayman El-Sayed. Impact of simple cheating in application-level multi-cast. In *INFOCOM*, 2004.
21. D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, 14, 1996.
22. Thomas Moscibroda, Stefan Schmid, and Roger Wattenhofer. On the topologies formed by selfish peers. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 133–142, New York, NY, USA, 2006. ACM Press.
23. Herve Moulin and Scott Shenker. Strategyproof sharing of submodular access costs: Budget balance versus efficiency. *Economic Theory*.
24. Roger B. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3), August 1977.
25. Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, September 1997.
26. Giovanni Neglia, Giuseppe Lo Presti, Honggang Zhang, , and Donald F. Towsley. A network formation game approach to study bittorrent tit-for-tat. In *EuroFGI International Conference on Network Control and Optimization*, June 2007.
27. Christos Papadimitriou. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753, New York, NY, USA, 2001. ACM Press.
28. P. D. Straffin. *Game Theory and Strategy*. Mathematical Association of America, Washington, DC, 1993.
29. Eva Tardos. Network Formation Games and the Potential Function Method. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
30. Honggang Zhang, Giovanni Neglia, Donald F. Towsley, and Giuseppe Lo Presti. On unstructured file sharing networks. In *INFOCOM*, pages 2189–2197, 2007.

Eigenvector based Reputation Measures

Konstantin Avrachenkov¹, Danil Nemirovsky¹, Son Kim Pham², Roberto G. Cascella³,
Roberto Battiti³, Mauro Brunato³

¹ Institut National de Recherche en Informatique et Automatique,
INRIA Sophia Antipolis,
F-06902 Sophia Antipolis, France

{k.avrachenkov, dnemirov}@sophia.inria.fr

² Department of Programming Technology
Applied Mathematics and Control Processes Faculty
Saint Petersburg State University

Saint Petersburg, Russia

sonsecure@yahoo.com.sg

³ Information Engineering and Computer Science
Department

University of Trento

I-38100 Trento, Italy

{cascella,battiti,brunato}@disi.unitn.it

Abstract. Trust and reputation are imperative for Internet-mediated service provision, electronic markets, document ranking systems, P2P networks and ad hoc networks. BIONETS is not an exception as it implements a communication paradigm based on nodes that form virtual communities in peer-to-peer ad hoc fashion. Reputation is a more objective notion and Trust is a more subjective one. Also, the estimation of trust is quite application-specific. Reputation is typically acquired over a long time interval, whereas Trust is based on a personal reflection before taking a decision to interact with another node. In other words, the reputation about a person or a thing is given by the community and the trust is a decision taken by an individual member of the community to rely on the other party.

The reputation value of a node can be calculated by aggregating the information pertaining to the history of the node itself. We must distinguish between two types of information: private observation and public observation. The former refers to direct experience of first-hand information and the latter to information that is publicly available.

Several functions can be used to aggregate reputation values starting from a simple average of the feedbacks to a majority rule approach. In the latter, if a majority of nodes send positive feedbacks, then the node is assumed to be trustworthy. Other functions count first for positive and negative ratings separately and, then, define the score as the difference between the two computed values.

Many reputation systems are based on graph centrality measures. A social/information network is typically represented by a directed/undirected/weighted graph. If the graph is directed, an edge from node A to node B signifies that node A recommends node B. Here we review distributed approaches to the graph-based centrality/reputation measures. The distributed approaches are particularly needed for large systems such as the WWW or P2P networks. Intuitively, graph centrality measures are based on the following two observations: (a) it is more likely that individuals will interact with friends of friends than with unknown parties. This is the so-called transitivity of trust; and (b) individuals are inclined to trust more somebody who is trusted by some of their friends with high reputation. As case studies of the graph-based reputation systems we shall take PageRank, TrustRank and EigenTrust. The other graph-based reputation measures appear to be modifications of the latter ones. In fact, since PageRank, TrustRank and EigenTrust are only different in the definition of the entries of the matrix representing reputation distribution among the nodes and in the definition of personalised vector defining restart random walk distribution, many algorithms designed for one reputation rank metric will work for the other eigenvector-based reputation rank metrics.

The BIONETS system consists of U-Nodes that travel and form ad hoc communities, named “islands”, and of T-Nodes that provide context information related to specific locations. These islands of connected nodes are dynamic in nature and might be temporarily not reachable from other nodes in the system.

In this settings, it is worth noticing that traditional approaches cannot be used in BIONETS as the nodes must deal with disconnected operation. Thus, new solutions are required to enable the fostering of cooperation and the formation of reputation values when nodes join communities that last for a short time. Given these constraints, the graph-based approach, and in particular the personalised method, is a viable solution that can be used in parallel to reputation management systems when the network is partitioned and local interactions of the nodes are frequent.

We present several approaches to compute the reputation value in a distributed way. The results presented in this paper target U-Nodes as they are more powerful entities that can compute the reputation of services, which is not part of this document. Specifically, here we survey available distributed approaches to the graph-based

reputation measures. Graph-based reputation measures can be viewed as random walks on directed weighted graphs whose edges represent interactions among peers. We classify the distributed approaches to graph-based reputation measures into three categories. The first category is based on asynchronous methods. The second category is based on the aggregation/decomposition methods. And the third category is based on the personalisation methods which use the information available locally. We survey in detail all three categories.

1 Introduction

Trust and reputation are imperative for Internet mediated service provision, electronic markets, document ranking systems, P2P networks and Ad Hoc networks. BIONETS is not an exception as it implements a communication paradigm based on nodes that form virtual communities in peer-to-peer ad hoc fashion. Firstly for the definition of suitable solutions in BIONETS it is important to distinguish clearly between the notions of trust and reputation. Following the works [17,27], we can define Trust as the extent to which one party is willing to depend on something or somebody with a feeling of relative security, even though negative consequences are possible. And we can define Reputation as what is generally said or believed about a node's or thing's character or standing. Thus, Reputation is a more objective notion and Trust is a more subjective one.

Reputation is typically acquired over a long time interval, whereas Trust is based on a personal reflection before taking a decision to interact with another node. In other words, the reputation about a person or a thing is given by the community and the trust is a decision taken by an individual member of the community to rely on the other party.

1.1 Estimation of trust

The estimation of trust is quite application specific. Examples of trust metrics for instance can be found in [17,20,27] and in references therein.

The trustworthiness of a node can be computed from the reputation values, that we refer as public available information, or a node can further combine reputation with its own personal experience. If we define the first hand information for node j computed at node x as opinion O_{xj} and the reports received from designated agents d , in general $d > 1$, R_{dj} can be combined as shown in the equation

$$T_{xj} = (1 - w_p)O_{xj} + w_p \frac{\sum_d R_{dj} \cdot C_{xd}}{\sum_d C_{xd}}, \quad (1)$$

where T_{xj} is the trust node x calculates and w_p is the parameter to make a trade off between the public and private information and where C_{xd} is the credibility of node d estimated by node x .

Sometimes, when a node is willing to access a service, it can compute the risk of the interaction but even if the other party is untrustworthy, it might decide to interact anyway. In this case, the trustworthiness of a node is the evaluation of the risk of a transaction.

1.2 Estimation of reputation

The reputation value of a node can be calculated by aggregating the information pertaining to the history of the node itself. We must distinguish between two types of information: private observation and public observation. The former refers to direct experience of first-hand information and the latter to information that is publicly available. The reputation value can be either the public or private information or a combination of the two.

Several functions can be used to aggregate reputation values starting from a simple average of the feedbacks to a majority rule approach. In the latter, if a majority of nodes send positive feedbacks, then the node is assumed to be trustworthy. Other functions count first for positive and negative ratings separately and, then, define the score as the difference between the two computed values. eBay uses a similar mechanism to compute reputation [42].

More complex mechanisms that count for positive and negative ratings are Bayesian systems [17,38,39]. The beta probability density function is used to determine the expected probability θ for a node to behave

well, as shown in equation (2). It is based on prior experience that is constantly updated by recomputing α and β as follows: $\alpha = p + 1$ and $\beta = n + 1$, where p and n are the number of positive and negative feedbacks respectively

$$\text{Beta}(\theta, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}, \quad (2)$$

where $\alpha, \beta > 0, 0 \leq \theta \leq 1$ and $\Gamma(\cdot)$ is the Gamma function. Other aggregation functions weigh each feedback F_{ij} issued by node i for the node j with other factors, e.g., the credibility of the reporter node C_{xi} estimated by node x , as defined in the following equation

$$R_{xj} = \frac{\sum_i F_{ij} \cdot C_{xi}}{\sum_i C_{xi}}. \quad (3)$$

Other values can be used to weigh reputation, like a *quality* value that scores the importance of the transaction that nodes are reporting. The quality value can be function of the number of transactions which the reputation value is based on or the importance of those transactions. This can be useful to distinguish reputation values that are based on very small number of transactions or on very minor transactions as reputation values, thus, computed may not be an adequate indicator of the reputation of a node.

Using a quality value protects the system from *milking* agents that build up their reputation by behaving honestly in many minor unimportant transactions and, then, behave dishonestly in a few large transactions. If all transactions have the same weight, an agent can successfully milk the system by choosing a few large transactions to behave dishonestly in. The concept of transaction quality has been discussed in literature [41,43].

The feedbacks can also be aged to assign higher weights to more recent experiences. This ensures that stale reputation information is aged appropriately and lower weight is given as reputation becomes older. The weight is a function of the time when the feedback is created. For instance, if the information is t time units old, then, the weight of that information is computed as $e^{-\gamma t}$ where γ is the ageing constant that denotes the rate at which information ages. Equation (4) shows an example of the computation of the reputation value for node j made by node x , where i counts for the number of feedbacks to aggregate

$$R_{xj} = \frac{\sum_i F_i e^{-\gamma t_i}}{\sum_i e^{-\gamma t_i}}. \quad (4)$$

Other aggregating functions are based on a trust graph built from direct interactions between entities [40,43]. Nodes are the vertices of the graph and the edges are the direct interactions. The weight w_e on the edge e rates the transactions. The reputation of a node j is calculated at node x by considering all the available paths from x to j and it can be expressed as shown in equation (5), where u are the intermediate nodes along the path between the source and the destination

$$R_{xj} = \sum_{e \in \text{incoming}(j)} w_e \cdot \frac{R_{uj}}{\sum_{f \in \text{incoming}(j)} R_{uf}}, \quad (5)$$

where R_u is the trustworthiness of the nodes that have interacted with node j and f gives the transactions that node u had with nodes connected to x . An algorithm can use this aggregation function recursively to determine the reputation values of all nodes in the system.

The choice of the best aggregation function and its implementation depend on the application context and on the desired output. The aggregated reputation value must be presented in a way that is useful to the consumers of this information. The aggregation service may output a binary value (trusted or not trusted), values on a discrete scale (say from 1 to 5 or $\{-1, 0, 1\}$) or on a continuous scale (0 to 1). A value of 0 means a node is not trustworthy and a value of 1 means a node is completely trustworthy.

1.3 Graph based reputation measure

The computation of reputation measures is less application specific. Many reputation systems are based on graph centrality measures. A social/information network is typically represented by a directed/undirected/weighted

graph. If the graph is directed, an edge from node A to node B signifies that node A recommends node B. Here we review distributed approaches to the graph based centrality/reputation measures. The distributed approaches are particularly needed for large systems such as WWW or P2P networks. Intuitively, graph centrality measures are based on the following two observations: (a) it is more likely that individuals will interact with friends of friends than with unknown parties. This is so-called transitivity of trust; and (b) individuals incline to trust more somebody who is trusted by some of their friends with high reputation. As case studies of the graph based reputation systems we shall take PageRank [30], TrustRank [10] and EigenTrust [20]. The other graph based reputation measures appear to be modifications of the latter ones.

PageRank [30] is one of the principle criteria according to which Google search engine ranks Web pages. The basic idea of PageRank algorithm is to use the hyper-links as indication that one Web page recommends another Web page. Also, PageRank can be interpreted as the frequency that a random surfer visits a Web page. Thus, PageRank reflects the popularity and reputation of a Web page. The formal definition of PageRank is as follows: Denote by n the total number of pages on the Web and define the $n \times n$ hyper-link matrix P as follows. Suppose that page i has $k > 0$ outgoing links. Then $p_{ij} = 1/k$ if j is one of the outgoing links and $p_{ij} = 0$ otherwise. If a page does not have outgoing links, we call it a dangling page, and the probability is spread among all pages of the Web with some distribution v , namely, $p_{ij} = v_j$. In order to make the hyper-link graph connected, it is assumed that a random surfer goes with some probability to an arbitrary Web page with the distribution v . Sometimes the distribution v is called personalised vector. In the standard PageRank formulation, this distribution is chosen to be uniform. Thus, the PageRank is defined as a stationary distribution of a Markov chain whose state space is the set of all Web pages, and the transition matrix is

$$G = cP + (1 - c)ev, \quad (6)$$

where e is a vector whose all entries are equal to one, $v = \frac{1}{n}e^T$, and $c \in (0, 1)$ is the probability of following a link on the page and not jumping to a random page (it is chosen by Google to be 0.85). The constant c is often referred to as a damping factor. The Google matrix G is stochastic, aperiodic, and irreducible, so there exists a unique row vector π such that

$$\pi G = \pi, \quad \pi e = 1. \quad (7)$$

The row vector π satisfying (7) is called a *PageRank vector*, or simply *PageRank*. There is an important modification of PageRank called TrustRank [10]. TrustRank adopts a special choice of personalised vector v to perform spam detection. Specifically, in TrustRank algorithm a set of pages called seeds is selected. The seeds are the pages judged by human experts as non-spam pages. It is argued that a small amount of pages should be judged to effectively distinguish spam pages from non-spam ones. This general idea is implemented by a particular choice of personalised vector v that is not an uniform probability distribution as in standard PageRank algorithm but a distribution allocating higher probability to seeds.

EigenTrust [20] is a reputation measure used in P2P networks with the aim to identify malicious peers and to exclude them from the network. Each peer i of the network stores the number of satisfactory transactions it has had with peer j , $sat(i, j)$ and the number of unsatisfactory transactions it has had with that peer j , $unsat(i, j)$. Then, the difference between the values s_{ij} is calculated

$$s_{ij} = sat(i, j) - unsat(i, j). \quad (8)$$

By composing matrix P in the following way

$$p_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}, \quad (9)$$

we can now apply the already familiar PageRank scheme. Personalisation vector v for P2P networks is defined in similar way as in TrustRank. There are some peers in a P2P networks that are known to be trustworthy. They are called “pre-trusted” peers. Assuming that we have t pre-trusted peers in the network vector v is defined as $v_i = \frac{1}{t}$ if i is a pre-trusted peer and $v_i = 0$ if i is not a pre-trusted peer.

There is a drawback of PageRank as a reputation measure. All outgoing links from a node provide equal contributions. However, in many applications one node can have worse or better experience when interacting with another node and consequently, the relations between two nodes can have different level of trust. This factor was taken into account in [20,36,37]. Namely, now the entries of matrix P are not simply defined as uniform distributions over the outgoing links, but represent levels of trust the node has in respect to his peers. Thus, we could regard the *TrustRank* measure as a random walk on a weighted graph. We would like to mention that the other works that study the reputation systems for P2P networks are [31,32,34]. The authors of [36,37] apply the graph based reputation measures to Semantic Web and the authors of [8] suggest to use the graph based reputation measures in mobile Ad Hoc networks.

Since PageRank, TrustRank and EigenTrust are only different in the definition of the entries of matrix P and in the definition of personalised vector v , many algorithms designed for one reputation rank metric will work for the other eigenvector based reputation rank metrics. Thus, in our survey, if an algorithm can be applied to either PageRank, TrustRank or EigenTrust, we simply denote the outcome of the algorithm as a *Rank vector*. In particular, to find the value of the Rank vector, it is often convenient to transform the eigenproblem based definition to an equivalent form of the linear system [26,22]:

$$\pi = \pi c P + (1 - c)v. \quad (10)$$

The structure of the chapter is organised as follows. In Section 2 we discuss the BIONETS system and we present the integration and application of the graph based approach. In Section 3 we review the asynchronous approaches to the graph based reputation measures. Then, in Section 4 we review the aggregation/decomposition approaches. In fact, the aggregation/decomposition approaches can be regarded as some limiting cases of the asynchronous approaches. However, the class of aggregation/decomposition approaches is large and it deserves a special section. Finally, in Section 5 we review the personalised approach to the graph based reputation measures. The personalised approach uses the information available locally. This is a natural approach to the reputation measures as the reputation discounts quickly in the chain of acquaintances and might not be easily accessible across the network.

2 Application of graph-based reputation in BIONETS

The BIONETS system consists of U-Nodes that travel and form ad-hoc communities, named “islands”, and T-Nodes that provide context information related to specific locations. These islands of connected nodes are dynamic by nature and might be temporarily not reachable from other nodes in the system. The work presented in [44] highlights the importance of reputation to provide provision trust among nodes and to function as fitness criteria for service provision in an autonomic system, like the one envisioned in BIONETS.

The application of reputation management systems in BIONETS is constrained by the networking functionalities of the components which implement a scheme proper of delay tolerant networks (DTNs) if they are not in the same island. The reachability of all the nodes and, as consequence, the retrieve of trust information of U-Nodes is affected by the communication network. To overcome these issues, several approaches have been investigated to analyse how the outcome of a transaction can be collected from the system and how reputation values can be disseminated [45].

In this settings, it is worth noticing that traditional approaches cannot be used in BIONETS as the nodes must deal with disconnected operations. Thus, new solutions are required to enable the foster of cooperation and the formation of reputation values when nodes join communities that last for short time. Given these constraints, the graph based approach, introduced in Subsection 1.3, and in particular the personalised approach discussed in Section 5, is a viable solution that can be used in parallel to reputation management systems when the network is partitioned and local interactions of the nodes are frequent.

In fact, this scheme only requires the collection of feedbacks as nodes can compute the trustworthiness of each single U-Node by exploiting information acquired through direct acquaintances. U-Nodes can create a temporary community and benefit at the same time from the definition of reputation to decide upon their transactions.

In particular, the application of the distributed approach can be used to mitigate the effect of disconnected networks where the nodes designated to store and to compute the reputation of other nodes cannot be contacted immediately. However, the drawback of this approach lays on the fact that it is not always possible to exploit initial personal acquaintances as the interactions between the same pairs of nodes are not frequent.

In BIONETS, we must deal with the high mobility of the nodes and the methods that exploit personal acquaintances must consider this issue for the computation of the reputation value. In some application, mobility has the advantage of helping in the creation of so-called trust chain [46], but in our case, the computation of reputation can lack samples to have a correct estimation of nodes trustworthiness. Moreover, the high churn rate in BIONETS requires that nodes compute the reputation value by using asynchronous approaches and the computation should be fast so that nodes can quickly evaluate other parties before leaving the community.

In the following sections, we present several approaches to compute the reputation value in a distributed way. The results presented in this survey target U-Nodes as they are more powerful entities that can compute the reputation of services.

3 Asynchronous approach

The most standard way for the computation of the Rank vector is the method of power iteration. Namely, in the power iteration method one just needs to iterate equation (10). Namely,

$$\pi^{(t+1)} = \pi^{(t)}cP + \frac{1-c}{n}e^T, \quad t = 0, 1, \dots, \quad (11)$$

with $\pi^{(0)} = \frac{1}{n}e^T$. Since the matrix cP is sub-stochastic, the algorithm converges. Furthermore, its convergence rate is bounded by c [13]. The number of FLOPS required to achieve the accuracy ε is equal to $\frac{\log \varepsilon}{\log c} \text{nnz}(P)$, where $\text{nnz}(P)$ is the number of nonzero elements of the matrix P [33]. Even though an implementation of the power iteration method for sparse matrices can be very efficient, one still would like to distribute its computation. The reasons for this are two-fold. Firstly, the computation on parallel computers can significantly accelerate the basic algorithm. In particular, one can apply GRID technology [9]. Secondly, in some applications like P2P network a distributed approach to the computation of reputation measures is indispensable. Below we review deterministic and stochastic approaches to the asynchronous computation of the graph based reputation measures.

The asynchronous iterations for the solution of fixed point linear systems like (10) was proposed in [6]. The class of asynchronous iterative methods of [6] can be described as follows:

$$\pi_j^{(t+1)} = \begin{cases} \sum_{i=1}^n cP_{ij}\pi_i^{(t-d(t,i,j))} + \frac{1-c}{n} & \text{if } j \in U(t), \\ \pi_j^{(t)} & \text{if } j \notin U(t), \end{cases} \quad (12)$$

where the function $U(t)$ gives a set of states to be updated at each step, and the function $d(t, i, j)$ gives the relative ‘‘age’’ of the entries used in the updates. Then, from [14,15] we have the following result about the convergence of asynchronous methods.

Theorem 1. *Let the functions $U(t)$ and $d(t, i, j)$ satisfy the following conditions:*

1. *Each vector entry, j , features in an infinite number of update sets;*
2. *For each pair of vector entries, i and j , we have that $(t - d(t, i, j)) \rightarrow \infty$ as $t \rightarrow \infty$ as well as $\forall t : d(t, i, j) \leq t$.*

Then, if the spectral radius of cP is strictly less than one, every sequence of iterates within the class given by (12) converges to the unique fixed point.

The authors of [21] have shown that the asynchronous iterates also converge in the eigenproblem formulation with the largest eigenvalue equal to one.

In P2P network a similar approach to asynchronous one is used. Each peer having requested trust values from other peers calculates its own value of trust

$$\pi_i^{(k+1)} = c(p_{1i}\pi_1^{(k)} + \dots + p_{ni}\pi_n^{(k)}) + (1 - c)v_i \tag{13}$$

and reports it to others.

Monte Carlo method provides a framework for the construction of stochastic asynchronous methods [2,5]. Let us for example describe one particular method from [2].

Algorithm 1 *Simulate N runs of the random walk initiated at a randomly chosen node. For any node j , evaluate π_j as the total number of visits to node j multiplied by $(1 - c)/N$.*

We note that the random walks are generated independently, which provides a natural framework for distributed implementation. Having information about a sub-graph ("island") of the whole graph at a node the random walks can be generated at the node until the random walk leaves the sub-graph. After that when it will be possible (two islands establish a connection) the generated parts of random walks will be merged producing estimation of Rank vector by Algorithm 1. As was shown in [2], to find nodes with high reputation it is enough to simulate the random walk a number of times equal to the number of nodes. This is in turn equivalent to the complexity of just one iteration of the power iteration method.

4 Aggregation/Decomposition Approach

Aggregation/decomposition methods (A/D methods) for computation of the Rank vector use the decomposition of the set of pages which we denote by I . Let us assume that the set I is decomposed into $N \leq n$ non-intersecting sets $I^{(i)}, i = 1, \dots, N$, such that

$$\begin{aligned} I^{(1)} &= \{1, \dots, n_1\}, \\ I^{(2)} &= \{n_1 + 1, \dots, n_1 + n_2\}, \\ &\vdots \\ I^{(N)} &= \{\sum_{i=1}^{N-1} n_i + 1, \dots, \sum_{i=1}^N n_i\}, \end{aligned} \tag{14}$$

with $\sum_{i=1}^N n_i = n$.

According to the decomposition of the set of pages the transition matrix can also be partitioned as follows:

$$P = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1N} \\ P_{21} & P_{22} & \dots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \dots & P_{NN} \end{pmatrix},$$

where P_{ij} is a block with dimension $n_i \times n_j$. In the same manner the Google matrix G can be presented in blocks,

$$G = \begin{pmatrix} G_{11} & G_{12} & \dots & G_{1N} \\ G_{21} & G_{22} & \dots & G_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \dots & G_{NN} \end{pmatrix}. \tag{15}$$

Following the partitioning of the Google matrix, the Rank vector is partitioned into components:

$$\pi = (\pi_1, \pi_2, \dots, \pi_N), \tag{16}$$

where π_i is a row vector with $dim(\pi_i) = n_i$. All aggregation methods use an aggregation matrix A . The matrix A is a matrix whose each element corresponds to a block of matrix G , i.e. $a_{ij} \leftrightarrow G_{ij}$. Typically the elements of the matrix A are formed as $a_{ij} = \zeta_i G_{ij} e$, where ζ_i is a probability distribution vector. We call

the vector ζ_i the *aggregation vector*. Each aggregation method forms the aggregation matrix in its own way using different probability distributions as aggregation vectors and different partitioning. One can consider the aggregated matrix as a transition matrix of a Markov chain with state space formed by sets of pages.

The convergence rate of an aggregation method depends on the choice of the decomposition. The aggregation method converges faster than power iteration method if off-diagonal blocks P_{ij} are close to zero matrix. It means that the random walk performed by the transition matrix G most likely stays inside sets $I^{(i)}$ and with small probability goes out.

In the following discussion the aggregation methods are applied to the Google matrix (6) and the Rank vector (7), but some of them can be applied to a general (irreducible or primitive) stochastic matrix and its stationary probability distribution.

4.1 Block-diagonal case

Let us consider the case when all blocks excluding the diagonal ones are zeroes [1], i.e.

$$P = \begin{pmatrix} P_1 & 0 & \dots & 0 \\ 0 & P_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_N \end{pmatrix}.$$

Since P is a stochastic matrix then all P_i are stochastic. For the i^{th} block define the Google matrix

$$G_i = cP_i + (1-c)(1/n_i)ee^T,$$

where the vector e has an appropriate dimension. Let vector π_i be the Rank vector of G_i ,

$$\pi_i = \pi_i G_i.$$

Then the original Rank vector π is expressed by

$$\pi = \left(\frac{n_1}{n} \pi_1, \frac{n_2}{n} \pi_2, \dots, \frac{n_N}{n} \pi_N \right).$$

The block-diagonal structure of the matrix P allows to produce computation of each component of the Rank vector in absolutely independent way from the other components.

4.2 Full aggregation method (FAM)

The method is based on the theory of stochastic complement and the coupling theorem [25]. Here we introduce it for the completeness.

Definition 1 (Stochastic complement). For a given index i , let G_i denote the principal block sub-matrix of G obtained by deleting the i^{th} row and i^{th} column of blocks from G , and let G_{i*} and G_{*i} designate

$$G_{i*} = (G_{i1} G_{i2} \dots G_{i,i-1} G_{i,i+1} \dots G_{iN})$$

and

$$G_{*i} = \begin{pmatrix} G_{1i} \\ \vdots \\ G_{i-1,i} \\ G_{i+1,i} \\ \vdots \\ G_{Ni} \end{pmatrix}.$$

That is, G_{i*} is the i^{th} row of blocks with G_{ii} removed, and G_{*i} is the i^{th} column of blocks with G_{ii} removed. The stochastic complement of G_{ii} in G is defined to be the matrix

$$S_i = G_{ii} + G_{i*} (I - G_i)^{-1} G_{*i}.$$

Theorem 2 ([25, Theorem 4.1] Coupling theorem). *The Rank vector of the Google matrix G partitioned as (15) is given by*

$$\pi = (v_1\sigma_1, v_2\sigma_2, \dots, v_N\sigma_N),$$

where σ_i are the unique stationary distribution vector for the stochastic complement

$$S_i = G_{ii} + G_{i*}(I - G_i)^{-1}G_{*i}$$

and where

$$v = (v_1, v_2, \dots, v_N)$$

is the unique stationary distribution vector for the aggregation matrix A whose entries are defined by

$$a_{ij} = \sigma_i G_{ij}e.$$

In respect to the scope of the survey, Theorem 2 is given in application to the Google matrix. For the most general formulation of the theorem an interested reader is referred to [25]. Theorem 2 implies that the Rank vector can be found by the exact aggregation but it forces to compute the stochastic complements of diagonal blocks and their stationary distributions. One can avoid it by using approximate iterative aggregation method.

Algorithm 2 *Determine an approximation $\pi^{(k)}$ to the Rank vector π of a Google matrix G in k iterations.*

1. Select a vector $\pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, \dots, \pi_N^{(0)})$ with $\pi^{(0)}e = 1$.

2. Do $k = 0, 1, 2 \dots$

(a) Normalise $\sigma_i^{(k)} = [\pi_i^{(k)}]$, $i = 1, \dots, N$.

(b) Form aggregated matrix $A^{(k)}$

$$a_{ij} = \sigma_i^{(k)} G_{ij}e.$$

(c) Determine the stationary distribution $v^{(k)}$ of $A^{(k)}$

$$v^{(k)} = v^{(k)} A^{(k)}.$$

(d) Determine disaggregated vector

$$\tilde{\pi}^{(k)} = (v_1^{(k)}\sigma_1^{(k)}, v_2^{(k)}\sigma_2^{(k)}, \dots, v_N^{(k)}\sigma_N^{(k)}).$$

(e) Do l steps of power iteration method

$$\pi^{(k+1)} = \tilde{\pi}^{(k)} G^l.$$

The Rank vector is the fixed point of Algorithm 2. Indeed, if $\pi^{(k)} = \pi$, then $A^{(k)} = A$ and $v^{(k)} = v$. Therefore, $\tilde{\pi}^{(k)} = \pi$, and $\pi^{(k+1)} = \pi$.

For the local convergence of the Algorithm 2 it is required to fulfil one of the conditions:

1. $G \gg 0$ and $G \geq \delta Q$, where $Q = e\pi$,
2. $G \geq eb$, where b is a row vector, $be = \delta$.

Since the Google matrix satisfies the both conditions, Algorithm 2 converges locally [24, Theorem 1]. Algorithm 2 also converges globally if l is large enough [24].

Let us provide an estimation of the rate of convergence of Algorithm 2 [28].

1. Consider the condition $G \geq \delta_1 Q$. Let us find δ_1 . Denote by g_{min} the minimum entry of the matrix G . If $p_{ij} = 0$ then $g_{ij} = g_{min}$. Hence,

$$g_{min} = \frac{1 - c}{n}. \tag{17}$$

The maximum of the elements of the Rank vector π is achieved when all the other elements achieving minimum, because of $\pi > 0$ and $\pi e = 1$. The minimum entry of the Rank vector for a page is realised

if there is no other page referring to it. The minimum entry of the Rank vector is $\frac{1-c}{n}$. Therefore the maximum of one of the element of the Rank vector is equal to

$$\pi_{max} = 1 - \frac{1-c}{n}(n-1) = \frac{1+c(n-1)}{n}. \quad (18)$$

Hence, if we find δ_1 from the constraint

$$g_{min} \geq \delta_1 \pi_{max}, \quad (19)$$

it ensures that $G \geq \delta_1 Q$. From the equalities (17), (18) and (19) we get

$$\delta_1 \leq \frac{1-c}{1+c(n-1)}.$$

2. Consider the condition $G \geq eb$, where $be = \delta_2$. Let us determine δ_2 . From the equalities (6) we obtain, that $G \geq \frac{1-c}{n}E$. The equality can be rewritten as $G \geq e \left(\frac{1-c}{n} e^T \right)$. Therefore, as the vector b one can take $\left(\frac{1-c}{n} e^T \right)$. Hence,

$$\delta_2 = 1 - c.$$

The error vector of the method at the k^{th} iteration is given by

$$\pi^{(k+1)} - \pi = (\pi^{(k)} - \pi)J(\pi^{(k)}).$$

The definition and expressions for the matrix $J(v)$ can be found in [24].

From the above estimation and [24] we can conclude that the spectral radius of the matrix $J(\pi)$:

1. is less than $1 - \delta_1 = \frac{cn}{1+c(n-1)} < 1$,
2. is less than $\sqrt{1 - \delta_2} = \sqrt{c} < 1$.

For n big enough the second estimation becomes better than the first one. The second estimation ensures that the convergence rate of the method is no less than \sqrt{c} . Unfortunately, the estimation does not ensure that the method converges faster than the power iteration method. Nevertheless, for the partial aggregation method which is discussed in the next subsection and is actually a particular case of the full aggregation method it was shown that there exists such partitioning of the Google matrix which provides faster convergence than the convergence of the power iteration method.

4.3 Partial aggregation method (PAM)

The partial aggregation method is considered in detail in [11]. Here we discuss the application of the method to the Google matrix and the Rank vector. The method is applied to the 2×2 case, i.e. $N = 2$, and the matrix G is partitioned as follows

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}.$$

The matrix $I - G$ is singular, but the matrix $I - G_{11}$ is non-singular [3]. Hence we can factor $I - G = LDU$ [25, proof of Theorem 2.3], where

$$\begin{aligned} L &= \begin{pmatrix} I & 0 \\ -G_{21}(I - G_{11})^{-1} & I \end{pmatrix}, \\ D &= \begin{pmatrix} I - G_{11} & 0 \\ 0 & I - S_2 \end{pmatrix}, \\ U &= \begin{pmatrix} I - (I - G_{11})^{-1}G_{12} \\ 0 & I \end{pmatrix}, \end{aligned}$$

and where S_2 is a stochastic complement of the block G_{22} .

Since the matrix U is non-singular we have $\pi(I - G) = 0$ if and only if $\pi LD = 0$. Hence

$$\pi_2 S_2 = \pi_2 \quad \pi_1 = \pi_2 G_{21}(I - G_{11})^{-1}, \tag{20}$$

which means that π_2 is a stationary distribution for the matrix S_2 . The expression (20) represents a particular case of Theorem 2 for the 2×2 decomposition of the transition matrix [25, Colorary 4.1]. The matrix S_2 has unique stationary distribution

$$\sigma_2 S_2 = \sigma_2, \quad \sigma_2 e = 1.$$

And we can find π_2 as $\pi_2 = \rho \sigma_2$, where the factor ρ is chosen to satisfy the normalisation condition $\pi e = 1$.

The component π_1 and the factor ρ can be expressed as components of the stationary distribution of the aggregated matrix

$$A_1 = \begin{pmatrix} G_{11} & G_{12}e \\ \sigma_2 G_{21} & \sigma_2 G_{22}e \end{pmatrix}.$$

From (20), $\pi_2 = \rho \sigma_2$ and $\sigma_2 e = 1$ we get

$$(\pi_1, \rho)(I - A_1) = 0, \quad (\pi_1, \rho)e = 1.$$

Since A_1 is stochastic and irreducible [25, Theorem 4.1], it has a unique stationary distribution α ,

$$\alpha A_1 = \alpha, \quad \alpha e = 1.$$

By the uniqueness we get $\alpha = (\pi_1, \rho)$.

The above analysis implies that the Rank vector can be found by the partial exact aggregation but it forces to compute the stochastic complement of G_{22} , block of the Google matrix G and its stationary distribution. One can avoid this by using the approximate iterative partial aggregation method.

Algorithm 3 Determine an approximation $\pi^{(k)}$ to the Rank vector π of a Google matrix G in k iterations.

1. Select a vector $\pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)})$ with $\pi^{(0)} e = 1$.
2. Do $k = 0, 1, 2, \dots$
 - (a) Normalise $\sigma_2^{(k)} = [\pi_2^{(k)}]$.
 - (b) Form aggregated matrix $A_1^{(k)}$

$$A_1^{(k)} = \begin{pmatrix} G_{11} & G_{12}e \\ \sigma_2^{(k)} G_{21} & \sigma_2^{(k)} G_{22}e \end{pmatrix}.$$

- (c) Determine the stationary distribution $\alpha^{(k)}$ of $A_1^{(k)}$

$$\alpha^{(k)} = \alpha^{(k)} A_1^{(k)}.$$

- (d) Partition $\alpha^{(k)}$

$$\alpha^{(k)} = (\omega_1^{(k)}, \rho^{(k)}).$$

- (e) Determine disaggregated vector

$$\tilde{\pi}^{(k)} = (\omega_1^{(k)}, \rho^{(k)} \sigma_2^{(k)}).$$

- (f) Do l steps of power iteration method

$$\pi^{(k+1)} = \tilde{\pi}^{(k)} G^l.$$

Let us consider $l = 1$. Algorithm 3 is the power iteration methods with matrix \tilde{G} [11, Proposition 5.1, Theorem 5.2], where

$$\tilde{G} = \begin{pmatrix} 0 & 0 \\ G_{21}(I - G_{11})^{-1} & S_2 \end{pmatrix}$$

Therefore, the rate of convergence of Algorithm 3 is equal to $|\lambda_2(S_2)|$ [11, Theorem 5.2]. If power iteration methods converges for matrix G then Algorithm 3 converges, too [11, Proposition 7.1]. If we consider a general stochastic matrix instead of the Google matrix Algorithm 3 can converge slower than the power iteration method [11, Example 6.3], but for the Google matrix there always exists such decomposition which ensures that Algorithm 3 converges faster than the power iteration method.

4.4 BlockRank Algorithm (BA)

The next method exploits the site structure of the Web. According to the experiments made by the authors of [18] the majority of links are the links between pages inside Web sites. Hence, we can decompose the set of pages I into the subsets according to the Web sites, i.e. $I^{(i)}$ is the set of the pages of site i . Then, the Google matrix is partitioned according to the decomposition of I .

Algorithm 4 Determine an approximation $\pi^{(k)}$ to the Rank vector π of the Google matrix G in k iterations.

1. Determine local Rank vector for each diagonal block P_i

- (a) Normalise P_i , i.e. $(\bar{P}_i)_{jk} = \frac{(P_i)_{jk}}{(P_i)_j \mathbf{1}}$.
- (b) Form G_i , $G_i = c\bar{P}_i + (1-c)(1/n)E$.
- (c) Approximately determine $\bar{\pi}_i$
 - i. Select a vector $\pi_i^{(0)}$.
 - ii. Do $k = 1, 2, \dots$

$$\pi_i^{(k)} = \pi_i^{(k-1)} G_i.$$

2. Determine BlockRank

- (a) Form aggregated matrix A
 - i. Do $k = 1, 2, \dots$
- (b) Form B , $B = cA + (1-c)(1/n)E$.
- (c) Approximately determine β
 - i. Select a vector $\beta^{(0)}$.
 - ii. Do $k = 1, 2, \dots$

$$a_{ij} = \bar{\pi}_i P_{ij} e.$$

$$\beta^{(k)} = \beta^{(k-1)} B.$$

3. Determine global Rank vector

- (a) Form the vector $\pi^{(0)} = (\beta_1 \bar{\pi}_1, \beta_2 \bar{\pi}_2, \dots, \beta_N \bar{\pi}_N)$.
- (b) Do $k = 1, 2, \dots$

$$\pi^{(k)} = \pi^{(k-1)} G.$$

It was empirically shown that Algorithm 4 is faster than the power iteration method by at least the factor of two [18].

4.5 Fast Two-Stage Algorithm (FTSA)

The next method also exploits the structure of the Web and in particular the presence of dangling nodes [23]. The main idea of the method is to lump dangling nodes into one state and find the Rank vector of the new aggregated matrix at the first stage and to aggregate non-dangling pages into one state at the second stage. Therefore, the set of pages is decomposed into two sets I_1 and I_2 , where $I_1 \cup I_2 = I$, and I_1 contains all non-dangling pages and I_2 contains all dangling pages. Hence, the matrix G is represented in the following way:

$$G = \begin{pmatrix} G_{11} & G_{12} \\ e_{n_1} v_{n_1} & e_{n_2} v_{n_2} \end{pmatrix},$$

where $e = (e_{n_1}^T, e_{n_2}^T)^T$ and $v = (v_{n_1}, v_{n_2})$.

Algorithm 5 Determine an approximation $\pi^{(k)}$ to the Rank vector π of the Google matrix G in k iterations.

1. The first stage: lump dangling pages

- (a) Form the lumped matrix $G^{(1)}$

$$G^{(1)} = \begin{pmatrix} G_{11} & G_{12} e_{n_2} \\ v_{n_1} & v_{n_2} e_{n_2} \end{pmatrix}.$$

(b) *Approximately determine π_1*

- i. Select a vector $\pi_1^{(0)}$.
- ii. Do $k = 1, 2, \dots$

$$\pi_1^{(k)} = \pi_1^{(k-1)} G^{(1)}.$$

(c) *Determine aggregation weights of the second stage*

$$\eta = \frac{\pi_1}{\sum_{i=1}^{n_1} (\pi_1)_i}.$$

2. *The second stage: aggregate non-dangling pages*

(a) *Form aggregated matrix $G^{(2)}$*

$$G^{(2)} = \begin{pmatrix} \eta G_{11} e_{n_1} & \eta G_{12} \\ v_{n_1} e_{n_1} & e_{n_2} v_{n_2} \end{pmatrix}.$$

(b) *Approximately determine π_2*

- i. Select a vector $\pi_2^{(0)}$.
- ii. Do $k = 1, 2, \dots$

$$\pi_2^{(k)} = \pi_2^{(k-1)} G^{(2)}.$$

3. *Form the Rank vector*

$$\pi = (\pi_1, \pi_2).$$

The first stage requiring less computation work than the power iteration method does, roughly $O(n_1)$ as opposed to $O(n)$ per iteration, and converges at least as fast as the power iteration method. The second stage usually converges to after about three iterations. If the second stage does not converge after about three iterations the acceleration based on Aitken Extrapolation [19] can be applied:

$$(\pi_2)_i = \frac{\left((\pi_2^{(2)})_i - (\pi_2^{(1)})_i \right)^2}{(\pi_2^{(3)})_i - 2(\pi_2^{(2)})_i + (\pi_2^{(1)})_i}.$$

4.6 Distributed PageRank Computation (DPC)

The following method is designed for distributed computation of the Rank vector [35]. The set of pages is decomposed by sites, i.e. $I^{(i)}$ is the pages of site i . The main idea of the method is to allow each site to compute the Rank vector for local pages and after that construct the entire Rank vector. (We refer to a site as a *super-node* which can independently perform computations.) After the Rank vectors for local pages is constructed by each super-node, the vectors are sent to selected central node (for example, central node can be one of the super-nodes), which constructs an aggregated matrix and determines its stationary distribution. The entries of the aggregated stationary distribution are delivered to super-nodes each of which constructs extended local transition matrix. Super-node determine stationary distribution of the extended local transition matrix and a part of the stationary distribution is normalised and reported to central node. The process is repeated until convergence. The formal definition of the algorithm is the following.

Let $S(\pi)$ denote a $N \times n$ disaggregation matrix as

$$S(\pi) = \begin{pmatrix} S(\pi)_1 & 0 & \dots & 0 \\ 0 & S(\pi)_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S(\pi)_N \end{pmatrix},$$

where $S(\pi)_i = [\pi_i]$ is a row vector denoting the censored stationary distribution of pages in site i . Also let us denote by G_i the i th block row of the matrix G partitioned as (15), i.e.

$$G_i = (G_{i1}, G_{i2}, \dots, G_{iN}).$$

Algorithm 6 Determine an approximation $\pi^{(k)}$ to the Rank vector π of the Google matrix G in k iterations.

1. For each super-node i

(a) Construct an $n_i \times n_i$ local transition matrix \bar{G}_{ii} , i.e. normalise G_{ii}

$$(\bar{G}_{ii})_{jk} = \frac{(G_{ii})_{jk}}{(G_{ii})_j \mathbf{1}}.$$

(b) Determine the stationary distribution $\pi_i^{(0)}$ for \bar{G}_{ii} .

2. Do $k = 0, 1, 2, \dots$

3. the central node

(a) Normalise $\sigma_i^{(k)} = [\pi_i^{(k)}]$.

(b) Construct the aggregated matrix $A^{(k)}$ with

$$a_{ij} = \sigma_i^{(k)} G_{ij} e.$$

(c) Determine the stationary distribution $\mathbf{v}^{(k)}$ for $A^{(k)}$.

4. For each super-node i

(a) Construct an $(n_i + 1) \times (n_i + 1)$ extended local transition matrix

$$B_i^{(k)} = \begin{pmatrix} G_{ii} & e^T (I - G_{ii}) \\ \frac{(v^{(k)} S(\pi^{(k)}))_{G_i - v_i^{(k)}} \pi_i^{(k)} G_{ii}}{1 - v_i^{(k)}} & \alpha^k \end{pmatrix},$$

where the scalar α^k ensures the row sum of $B_i^{(k)}$ is one.

(b) Determine the stationary distribution b_i^k for $B_i^{(k)}$.

(c) Partition b_i^k

$$b_i^k = (\omega_i^{(k)}, \beta_i^{(k)}),$$

where $\beta_i^{(k)}$ is a scalar.

(d) Form local vector $\tilde{\pi}_i^{(k)}$

$$\tilde{\pi}_i^{(k)} = \frac{1 - v_i^{(k)}}{\beta_i^{(k)}} \omega_i^{(k)}.$$

5. The central node

(a) Construct vector $\tilde{\pi}^{(k)}$

$$\tilde{\pi}^{(k)} = \left(\tilde{\pi}_1^{(k)}, \tilde{\pi}_2^{(k)}, \dots, \tilde{\pi}_N^{(k)} \right).$$

(b) Normalise $\pi^{(k)} = [\tilde{\pi}^{(k)}]$.

The method is proved to be equivalent to iterative aggregation/disaggregation method based on Block Jordan decomposition [35]. The main advantage of this method is that it provides a distributed way to calculate the Rank vector, at the same time the communication overhead is not high due to only scalars and vector are being sent between nodes and not matrices. The entire communication overhead is of the magnitude $O(nnz((\bar{L} + \bar{U})S(\pi))) + O(n)$, where \bar{L} and \bar{U} are low-triangular and upper-triangular matrices of the matrix P , respectively.

4.7 Discussion of the A/D methods

The considered A/D methods can be classified into two groups. While FAM and PAM are general methods which can be applied to any decomposition of the set of nodes, BA and FTSA propose to use a particular decomposition. The rate of convergence of FAM and PAM essentially depends on the chosen decomposition. The questions of ‘‘optimal’’ decomposition of the set of nodes and convergence acceleration are still not fully answered. BA and FTSA methods have made a step in the direction of solving the both questions. Whilst the BA method accelerates the power iteration method, the FTSA method converging as fast as the power iteration method reduces the dimension of matrices and vectors used in iterations. The DPC method, being distributed and parallel method, converges like aggregation-disaggregation method and at the same time has low communication overhead.

5 Personalised approach

The above presented algorithms compute the global reputation measure. Namely, to calculate the Rank vector we need an input (may be indirect) from all the nodes. However, the reputation discounts quickly in the chain of acquaintances. This provides a motivation to consider “localised” or “personalised” versions of the graph based reputation measures. Furthermore, one often needs to encompass different notions of importance or reputation for different users and queries. Thus, the original algorithm should be modified to take into account personalised view for the reputation of the nodes.

As we mentioned in the introduction, in general a random walk follows the outgoing links with probability c , and makes a random jump with probability $(1 - c)$ according to the probability distribution given in v . Depending on the “type” of users, vector v will not be uniform, but biased to some set of nodes, which are considered to be important for these “types” of users. For this reason, the vector v is referred as *personalisation* vector. Let $\pi(v)$ denote the personalised Rank vector (PRV), corresponding to the personalisation vector v . It can be computed by solving the equation $\pi = \pi G$

$$\begin{aligned}\pi &= \pi cP + (1 - c)v, \\ \pi - \pi cP &= (1 - c)v, \\ \pi(I - cP) &= (1 - c)v.\end{aligned}$$

Since c is different from one, the matrix $(I - cP)$ is invertible and we have

$$\pi(v) = v(1 - c)(I - cP)^{-1}. \quad (21)$$

Let $Q = (1 - c)(I - cP)^{-1}$. By letting $v = e_i^T$ ⁴ we see that $\pi(e_i) = Q^i$ – the i^{th} row of Q . Thus, rows of Q comprise a complete basis for personalised Rank vectors. Any PRV can be expressed as a convex linear combination of these basic vectors. This statement is based on the following theorem:

Theorem 3. *Given two arbitrary π_1, π_2 PRVs and v_1, v_2 are their corresponding personalisation vectors. Then, for any constants $\alpha_1, \alpha_2 > 0$ such that $\alpha_1 + \alpha_2 = 1$*

$$\alpha_1 \pi_1 + \alpha_2 \pi_2 = c(\alpha_1 \pi_1 + \alpha_2 \pi_2)P + (1 - c)(\alpha_1 v_1 + \alpha_2 v_2) \quad (22)$$

For any personalisation vector v , the corresponding PRV is given by vQ . Unfortunately, approach to use the complete basis for the personalised Rank vector is infeasible in practice. Computing the dense matrix Q off line is impractical due to its huge size. However, rather than using the full basis, we can use a reduce basis with $k < n$ vectors. In this case, we can not express all PRVs but only those corresponding to convex combinations of the vectors in reduced basis set

$$\pi(\omega) = w\hat{Q}. \quad (23)$$

5.1 Scaled Personalisation

In [16] the authors have presented a method that enables the computation of PRVs which scales well with the increasing number of users. The authors of [16] have developed their method in the context of information retrieval. Then, the authors of [7] have adopted the method of [16] to the reputation management in P2P networks. In this survey, we also present a broader reputation measure based interpretation of the algorithm of [16]. We would like to mention that the division of the users in two groups: pre-trusted peers and regular users provide yet another application of the results of [7,16] in the context of BIONETS [4].

The central notion of the scaled personalisation algorithm is the set of pre-trusted peers (or hub peers). A regular peer can choose some of pre-trusted peers. This does not mean that the hub peers selected by a user more trustworthy than the other hub peers. This simply means that a user might prefer certain hub peers because they supply a specific service of a very good quality. Next let us describe several stages of the scaled personalisation algorithm.

⁴ e_i has 1 in i^{th} place, and 0 elsewhere.

Specification Let us consider a set of the personalisation vectors u_h where $u_h = e_h$ is biased to a specific hub node $h \in H$. We denote by H the set of hub nodes. The personalised Rank vector corresponding to u_h is called a basis hub vector π_h . If the basis vector for each hub node $h \in H$ is computed and stored, then, by Theorem 3 any PRV corresponding to a preference set $P \subseteq H$ can be computed. The preference set P corresponds to the set of hub nodes chosen by a user as preferred pre-trusted peers.

Each hub vector can be computed naively by power method. However, this task is very expensive in time and resources. The algorithm of [16] enables a more scalable computation by constructing hub vectors from shared components.

Decomposition of Basis Vectors To compute a large number of basis hub vectors efficiently, one can decompose them into partial vectors and hubs skeleton, components from which hub vectors can be constructed quickly.

Let define the *inverse P-distance* $r'_p(q)$ from p to q as

$$r'_p(q) = \sum_{t:p \rightsquigarrow q} P[t](1-c)c^{l(t)}, \quad (24)$$

where the summation is taken over all tours t , starting from p and finishing at q , possibly visiting p and q more than one time, $l(t)$ is the length of the tour, and $P[t]$ is the probability of taking the tour t .

Consider tour $t = \langle w_1, \dots, w_k \rangle$, then

$$P[t] = \prod_{i=1}^{k-1} \frac{1}{\text{Outdeg}(w_i)},$$

or 1, if $l(t) = 0$. If there is no any tours from p to q , the summation is taken to be equal to 0. It is proven that $\pi_p(q) = r_p(q)$ [16].

Let us also define the *restricted inverse P-distance*. Let $H \subseteq V$ be some nonempty set of nodes. For $p, q \in V$, $r_p^H(q)$ is defined as a restriction of $r_p(q)$ that considers only tours from p to q that pass through H , that is,

$$r_p^H(q) = \sum_{t:p \rightsquigarrow H \rightsquigarrow q} P[t](1-c)c^{l(t)}. \quad (25)$$

Intuitively, $r_p^H(q)$ is the influence of p on q through H . Obviously, if all paths from $p \rightsquigarrow q$ come through H , then $r_p^H(q) = r_p(q)$. For carefully chosen H , $r_p(q) - r_p^H(q) = 0$ for many pages p, q . The strategy is to take advantage of this property by breaking r_p into components $(r_p - r_p^H)$ and r_p^H .

$$\pi_p = r_p = (r_p - r_p^H) + r_p^H. \quad (26)$$

The vector $(r_p - r_p^H)$ is called the *partial vector*. Computing and storing partial vectors is cheaper, since they can be represented as a list of their nonzero entries. Moreover, the size of each partial vector will decrease as H increases in size, making this approach particularly scalable. It can be proven that any r_p^H vector can be expressed in terms of the partial vectors $(r_h - r_h^H)$ for $h \in H$ (see the Hub Theorem in [16]).

Theorem 4. For any $p \in V, H \subseteq V$,

$$r_p^H = \frac{1}{1-c} \sum_{h \in H} (r_p(h) - (1-c)x_p(h))(r_h - r_h^H - (1-c)x_h), \quad (27)$$

where $x_h = e_h$. The quantity $(r_h - r_h^H)$ appears on the right hand side of (27) is the partial vector. Suppose we have computed $r_p(H) = \{(h, r_p(h)) | h \in H\}$ for a hub node p . Substituting it into equation (26) gives

$$r_p = (r_p - r_p^H) + \frac{1}{1-c} \sum_{h \in H} (r_p(h) - (1-c)x_p(h))[(r_h - r_h^H) - (1-c)x_h]. \quad (28)$$

The equation is central to the construction of hub vectors from partial vectors. The set $S = \{r_p(H) | p \in H\}$ forms the hubs skeleton, giving the interrelationships among partial vectors. Computing $(r_p - r_p^H)$, $p \in H$ naively by power method is inefficient due to the large number of hub nodes. Three scalable algorithms for computing these partial vectors, using dynamic programming are presented. All of them are based on the decomposition theorem in [16].

Theorem 5. For any $p \in V$

$$r_p = \frac{c}{|O(p)|} \sum_{i=1}^{|O(p)|} r_{O_i(p)} + (1-c)x_p \quad (29)$$

where $O_i(p)$ is the i^{th} neighbour of node p .

The above theorem gives the interpretation for PRV. The p 's view of r_p is the average of the views of its out-neighbours, but with extra importance given to p itself.

Construction of PRV's Let $u = \alpha_1 p_1 + \dots + \alpha_z p_z$ be a preference vector, where $p_i \in H$. Let

$$r_u(h) = \sum_{i=1}^z \alpha_i (r_{p_i}(h) - c \cdot x_{p_i}(h)). \quad (30)$$

Then, the PRV π can be computed as follows:

$$\pi = \sum_{i=1}^z \alpha_i (r_{p_i} - r_{p_i}^H) + \frac{1}{1-c} \sum_{h \in H} r_u(h) [(r_h - r_h^H) - (1-c)x_h]. \quad (31)$$

The choice of H The choice of hub nodes can have a strong effect on the overall performance. Particularly, the size of partial vectors is smaller when pages in H have high Rank vector values, since nodes with high Rank vector values are closer in term of P-reverse distance to other pages. In the context of P2P networks, it is natural for the members in the pre-trusted peers to have high Rank values.

5.2 Relation to the A/D approach

Let us relate the Personalised Rank vector approach to the A/D approach. The BlockRank algorithm proposed in [18] computes $n \times k$ matrix corresponding to k blocks. Each block corresponds to a host. Instead of choosing an uniform distribution over pages to which the user jumps, we may choose a distribution centred on hosts. So, we can encode the personalisation vector in the k -dimensional space. With this adaptation the local Rank vector will not change for different personalisation's. Only the BlockRank depends on the personalisation's. Therefore, we only need to recompute the BlockRank for each block-personalisation vector. The BlockRank algorithm is able to exploit the graph's block structure to compute efficiently many of the block-oriented basis vectors.

6 Conclusions

In this paper we reviewed reputation and trust measures that one can apply to BIONETS. Graph based reputation measures are discussed in details. Although all the methods can be effectively implemented in BIONETS environments, some of them has evident advantages and disadvantages. Aggregation/disaggregation methods like FAM, PAM, BA, DPC construct an aggregated matrix and its stationary distribution at each iteration of the algorithms that requires communication with selected central node. In application to BIONETS it means that before the iterations can be continued all the "islands" should connect to a selected node (which means there should be a node travelling to the selected nodes and back to an "island") to upload current state and, after a while, connect the same selected node to download aggregated distribution. This constraint is very stringent for BIONETS systems since it requires existence of nodes with special mobility pattern. Besides the mentioned disadvantages BA and DPC has an advantage over FAM, PAM and FTSA since BA and DPC calculate local Rank vectors as an initialisation to iterations that can be used as reputation before islands get a connection to the selected node and determine the

global reputation. The asynchronous approach and Monte Carlo method take an advantage comparing to aggregation/disaggregation methods since a lot of calculation of reputation can be done inside an island and the information of local reputation can be spread to other islands by nodes travelling occasionally from one island to another. While the spreading of information is achieved by merging generated random walks in Monte Carlo method, in asynchronous approach it is done by choosing appropriate functions of $U(t)$ and $d(t, i, j)$, where $U(t)$ is responsible for reachable at the moment nodes and $d(t, i, j)$ defines relative age of reputation values used in further calculations. Personalised approach is a kind of auxiliary approach to all the discussed methods because it chooses personalisation vector that can be used in aggregation/disaggregation methods as well as in the asynchronous approach and the Monte Carlo method. The personalised approach allows one to take into account preferences like special services provided by nodes or extremely high quality of common services.

References

1. K. Avrachenkov and N. Litvak. "Decomposition of the Google PageRank and Optimal Linking Strategy". *INRIA Research Report no. 5101*, 2004.
2. K. Avrachenkov, N. Litvak, D. Nemirovsky and N. Osipova, "Monte Carlo methods in PageRank computation: When one iteration is sufficient", *SIAM Journal on Numerical Analysis*, v.45, no.2, pp.890-904, 2007.
3. A. Berman and R.J. Plemmons. "Nonnegative Matrices in the Mathematical Sciences". *SIAM Classics In Applied Mathematics*, SIAM, Philadelphia, 1994.
4. Biologically inspired Network and Services (BIONETS): <http://www.bionets.org/>
5. L.A. Breyer, "Markovian Page Ranking Distributions: Some Theory and Simulations", *Technical report*, 2002; available online at <http://www.lbreyer.com/preprints.html>.
6. D. Chazan and W.L. Miranker, "Chaotic relaxation", *Linear Algebra and its Applications*, v.2, pp.199-222, 1969.
7. P.A. Chirita, W. Nejdl, M. Schlosser, and O. Scurtu, "Personalized reputation management in P2P networks", in *Proceedings of the Trust, Security, and Reputation Workshop*, 2004.
8. L. Eschenauer, V. Gligor and J. Baras, "On trust establishment in mobile Ad Hoc networks", in *Proceedings of Security Protocols Workshop*, pp.47-66, 2002.
9. I. Foster and C. Kesselman, (eds.) "The GRID: Blueprint for a new computing infrastructure", *Elsevier, San Francisco*, 2004.
10. Z. Gyongyi, and H. Garcia-Molina and J. Pedersen, "Combating Web Spam with TrustRank", *VLDB, Toronto, Canada*, 2004.
11. C.F. Ipsen and S. Kirkklad. "Convergence analysis of an improved PageRank algorithm". *NCSU CRSC Technical Report*, 2004.
12. T.H. Haveliwala, "Topic-Sensitive PageRank", in *Proceedings of the 11th International World Wide Web Conference*, 2002.
13. T.H. Haveliwala and S.D. Kamvar, "The Second Eigenvalue of the Google Matrix", *Stanford University Technical Report*, March 2003.
14. D. de Jager, "PageRank: Three distributed algorithms", *Master thesis, Imperial College (University of London)*, 2004.
15. D. de Jager and J.T. Bradley, "Asynchronous iterative solution for state-based performance metrics", in *Proceedings of ACM SIGMETRICS 2007*.
16. G. Jeh and J. Widom, "Scaling Personalized Web Search", in *Proceedings of the 12th International World Wide Web Conference*, 2003.
17. A. Josang, R. Ismail and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decision Support Systems*, v.43, no.2, pp.618-644, 2007.
18. S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. "Exploiting the block structure of the web for computing pagerank". *Stanford University Technical Report*, 2003.
19. S.D. Kamvar, T.H. Haveliwala, C.D. Manning, and G.H. Golub, "Extrapolation methods for accelerating PageRank computations", in *Proceedings of the 12th International World Wide Web Conference*, 2003.
20. S.D. Kamvar, M.T. Schlosser and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks", in *Proceedings of the 12th International World Wide Web Conference*, 2003.
21. G. Kollias, E. Gallopoulos and D. Szyld, "Asynchronous iterative computations with Web information retrieval structures: The PageRank case", in *Proceedings of the International Conference ParCo 2005. Parallel Computing: Current & Future Issues of High-End Computing*, pp.309-316.
22. A.N. Langville and C.D. Meyer. "Deeper inside pagerank". *Internet Mathematics*, 1(3):335-380, 2005.
23. Ch.P.Ch. Lee, G.H. Golub, and S.A. Zenios, "A fast two-stage algorithms for computing PageRank", *SCCM Report*, 2002.
24. I. Marek and I. Pultarova. "Two notes on local and global convergence analysis of iterative aggregation-disaggregation method" *available on the Web*, 2005.
25. C.D. Meyer and R.J. Plemmons. "Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems". *SIAM Rev.*, 1989.
26. C.B. Moler. "Numerical Computing with MATLAB". *SIAM*, 2004.
27. L. Mui, M. Mohtashemi and A. Halberstadt, "A computational model of trust and reputation", in *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

28. D.A. Nemirovsky, "Analysis of iterative methods for PageRank computation based on decomposition of the Web graph", *Master thesis, St.Petersburg State University, in Russian*, 2005.
29. The Open Directory Project www.dmoz.org
30. L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: Bringing order to the Web", *Technical Report, Stanford Digital Library Technologies Project*, 1998.
31. K. Sankaralingam, S. Sethumadhavan and J.C. Browne, "Distributed pagerank for P2P systems", in *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, 2003.
32. S.M. Shi, J. Yu, G.W. Yang and D.X. Wang, "Distributed page ranking in structured P2P networks", in *Proceeding of International Conference on Parallel Processing*, 2003.
33. W.J. Stewart. "Introduction to the numerical solutions of Markov chains". *Princeton University Press, Princeton*, 1994.
34. A. Yamamoto, D. Asahara, T. Ito, S. Tanaka and T. Suda, "Distributed pagerank: a distributed reputation model for open peer-to-peer network", in *Proceedings of the International Symposium on Applications and the Internet Workshops*, 2004.
35. Y. Zhu and Sh. Ye and X. Li, "Distributed PageRank computation based on iterative aggregation-disaggregation methods", in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, an updated version is available at http://www.cs.cmu.edu/~yangboz/cikm05_pagerank.pdf.
36. C.-N. Ziegler and G. Lausen, "Spreading activation models for trust propagation", in *Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service*, 2004.
37. C.-N. Ziegler and G. Lausen, "Propagation models for trust and distrust in social networks", *Information Systems Frontiers*, v.7, no.4/5, pp.337-358, 2005.
38. S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226-236, Lausanne, Switzerland, 2002.
39. S. Buchegger and J.-Y. L. Boudec. A robust reputation system for P2P and mobile ad-hoc networks. In *Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge, MA, USA, 2004.
40. Z. Despotovic and K. Aberer. P2P reputation management: probabilistic estimation vs. social networks. *Computer Networks*, 50(4):485-500, 2006.
41. A. Garg, R. Battiti, and G. Costanzi. Dynamic Self-management of Autonomic Systems: The Reputation, Quality and Credibility (RQC) scheme. In *The 1st IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC 2004) (LNCS 3457)*, pages 165-176, Berlin, Germany, Oct. 2004. Springer.
42. P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. the economics of the internet and e-commerce. In M. R. Baye, editor, *Advances in Applied Microeconomics*, volume 11, pages 127-157. Elsevier Science, 2002.
43. L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust in peer-to-peer communities. *IEEE Transactions on Data and Knowledge Engineering, Special Issue on Peer-to-Peer Based Data Management*, 16(7):843-857, July 2004.
44. R. Cascella and A. Garg (Eds.). Trust and reputation management system definition. BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D4.1), June 2007.
45. D. Schreckling (Eds.). Towards security in BIONETS. BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D4.2), August 2007.
46. S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing*, 5(1):43-51, January 2006.

Historical-Interpretive Considerations about Money as a Unit of Value and Scale-Dependent Phenomenon

Silvia Elaluf-Calderwood

Department of Media and Communications
London School of Economics and Political Science
London, United Kingdom
S.M.Elaluf-Calderwood@lse.ac.uk

Abstract. The telecommunications industry has successfully adopted traditional business models for the dissemination of mobile technology based on a hierarchical distribution network of media content (voice and data). BIONETS is an innovative project, EU sponsored, which enables node based and decentralised distribution of content, where the idea of propagation within nodes is bio-inspired. This node-based network layout is an opportunity for the introduction and exploration of alternative economic models for content distribution. These models are already used in other commercial sectors as the basis for development of decentralised and parallel (but not substitute) business models for trade or exchange of goods (content). The paper presents a number of working cases in which these models are applied and the economic definitions used to define the operational environment. The aim of the paper is to introduce to the reader potential ways to use these models on the BIONETS network layout.

1 Introduction

This document aims to present a sectional overview of the social, economic and technological aspects that need to be taken into account when understanding the potential application for the Economics of Sharing and/or Community Currencies socio-economic models in the BIONETS environment. The social paradigm in use is presented as an alternative model parallel to current economic models and does not aim to be used as the only or exclusive model for development. Historically, the models to be discussed have been actively used for a long time but, since economic interconnections tend to be of a socio-geographical nature [1], the adoption of such models has been highly ideologically motivated, as a solution to social problems particular to a geographical location. Some of these models have a successful implementation and use, for example, the Local Exchange and Trading Systems (LETS) [2]. Since membership in these types of economic initiatives has been exclusive or partial and not economically motivated, they can be sustained by considerable social effort in the medium and long term [1].

In the case of BIONETS, the social paradigm for the development of economic models leading to a structure of alternative business models is being explored as part of the overall trends in research into mobile technology. The rise of mobile technology and the multiple applications and services that can be provided directly to users offer the opportunity for a liberal use of the economics of sharing and community currencies ideas, allowing their users to create a wider user base where exchanges are economically motivated. Ubiquitous technology such as mobile devices raises new possible applications in a virtual world aimed at understanding, applying and negotiating these concepts. This is a new area of social interaction being researched by social scientists. If technology increases the ability of people to share, the question still pending is whether they will share more than just those goods and services that can be exchanged by means of technology? [3].

This paper is structured in five sections. The first section presents a summary of the common idea of money as a phenomenon that is historically interpretative. The second and third sections reflect upon the meaning and ideological abstractions linked to the concepts of community currencies and the economics of sharing applied to current mobile technology models. The fourth section covers a further in-depth analysis of mobile technology as a potential new economic agent-mediator as well as some business

examples of how mobile users, and in general IT users, are using technology to establish new socio-economic money trade-offs. Finally, some words about further research in the area using the BIONETS infrastructure at the application layer and some convergent issues on sustainability end this paper.

2 An Interpretative Historical Background about Money

In this section a general discussion on the value of the models to be presented in later sections is linked to the idea of money. Money has a social interpretive value, as is clear if seeking a definition of money using neo-classical economics. In this document money is defined as a common ‘language’, in contrast to the economics field, in which there are several definitions of money; some of these definitions are overlapping while others have distinct semantic boundaries. In this text, the discussion of money is limited to the meaning it has acquired through a consensual process, highlighting the social interpretation of the idea of money as currency over other categorisations, whilst taking into account monetary theory for the understanding of the microeconomics in place. Thus, *money is commonly defined by the functions attached to any good or token that functions in trade as a medium of exchange, store of value, and unit of account* [4].

In everyday use money refers more specifically to currency, and to the circulation of currencies with a legal tender status, which works as a standard of deferred payment. Over the history of mankind commodity money systems were the first to appear; exchanges in salt, iron, copper, gold, silver, precious woods, etc. were used for trade, firstly between individuals, then social groups, and finally communities, tribes or nations. A main characteristic of commodity money is the fact that, in addition to the value the money mediates, there is an intrinsic value to the currency being traded; for example, wood can be used to build buildings besides being traded. This type of money value differs from the barter model in which goods are given equal value for trade; or even from the socially-based, multiple-commodity currency used by medieval towns in the Hanseatic league in the Northern areas of Europe [5].

Money and its benefits are socially discussed in terms of inflation and interest. Inflation is an expansion of the money supply and/or credit, usually resulting in an increase in prices (the symptom of inflation) because the currency devalues [6]. Economic theory uses a theoretical framework in which the neutrality of money is axiomatic and in which a change in the stock of money affects only nominal variables in the economy such as prices, wages and exchange rates, having no effect on real variables like GDP, employment, and consumption; these factors are taken as externalities by the economic models to be presented. In modern barter models these concepts deserve a more in-depth analysis in light of the fact that the barter model is used as the reference model of choice by modern community currencies such as LETS. The barter model encourages people to exchange goods and services within their local communities, creating an alternative economy outside, and parallel to, the wider economy to which the community might belong. The barter economy was highly developed during Medieval times for practical reasons: the lack of readily available gold currency for trade made this model a common practice, since it relies on reciprocity, distribution, and trust for its sustainability.

The metaphor to illustrate the idea of the bartering model is the medieval Sunday Village Fair, as illustrated in Figure 1.¹ The European Medieval village is a rich source of metaphors for community currencies and economics of sharing. All members of the village joined the common area and exchanged goods, news and other items. It is known that visitor traders from outside the village would have presented some form of accreditation to the local authorities, before proceeding to trade. The barter model would have been the choice between the members of the community as it would have coexisted in parallel with currency brought to the community by visitor traders.

In the Medieval Village there is limited centralisation when issuing money or currency as the medium for trade. There is trust between villagers or group of villagers at trading times for goods or services. The shift in money priorities between the feudal and capitalist systems can be seen from the contrasting modern description of money. Based on the common understanding of the phenomenon, this is commonly identified with Fiat Money, in which a central authority such as a national government creates a new

¹ This painting by Dutch painter Pieter Bruegel the Elder (1525-1569) is actually called “Children’s Games”, but to our modern eyes still evokes the bustle of a medieval village market (<http://gardenofpraise.com/art28.htm>).



Fig. 1. The Medieval Village Market Metaphor: Barter economy and multiple community currencies in action

money object that has negligible inherent value. The widespread acceptance of Fiat Money is most frequent when the central authority mandates the money's acceptance under penalty of law and demands this money in payment of taxes or tribute.

According to this centralised view, money needs to be provided with some essential value characteristics. The descriptive characteristics are [5]:

- It is a medium of exchange. A medium of exchange is an intermediary used in trade to avoid the inconveniences of a pure barter system.
- It is a unit of account. A unit of account is a standard numerical unit of measurement of the market value of goods, services, and other transactions.
- It is a store of value. To act as a store of value, a commodity or financial capital, money must be reliably saved, stored, and retrieved, and be predictably useful when it is so retrieved.

To acquire these characteristics, Fiat Money should have a number of other features [7], which are presented on Table 1. This table provides a general summary of the terms used to define money in this paper. The reader needs to take into account that money as we know it today (after the Bretton Woods Agreement of 1944) is a central piece of the credit-based monetary system. In this paper the interpretative analysis focuses on the use of money as part of the macro-theories applied to communities of trade, in which this definition of money is focused on expressing money as a numerical value and scale-dependent phenomenon, a generalisation of its use as the primary medium for transaction, ignoring potential aspects of money as a social exchange trading tool. It is difficult to envisage how the use of money as defined above can contribute to social community construction. The more this idea of money has taken over the world, the more we see that instead of creating wealth our money system is depleting our real wealth: our communities, ecosystems, and productive infrastructure (Korten, 1997). The time has come to review how we measure wealth and to look at alternatives such as Economics of Sharing and Community Currencies for strong conceptual and practical ways to increase wealth without such depletion, which Korten regards as unsustainable from the point of view of long-term benefits to humanity.

Enhancing economic power in communities in which money is relatively important but not essential for everyday activities might demand a definition of money in which there is a strong bias towards

Money as a medium of exchange	Money as a unit of account	Money as a store of value
1. It should have liquidity, and be easily tradeable, with a low spread between the prices to buy and sell; in other words, a low transaction cost.	1. It should be divisible into small units without destroying its value; precious metals can be coined from bars, or melted down into bars again. This is why leather, or live animals, are not suitable as money.	1. It should be long-lasting, durable; it must not be perishable or subject to decay. This is why food items, expensive spices, or even fine silks or oriental rugs, are not generally suitable as money.
2. It should be easily transportable; precious metals have a high value to weight ratio. This is why oil, coal, vermiculite, or water are not suitable as money even though they are valuable. Paper notes have proved highly convenient in this regard.	2. It should be fungible: that is, one unit or piece must be exactly equivalent to another.	2. It should have a stable value.
3. It should be physically durable and non-health hazardous.	3. It must be a specific weight, or measure, or size to be verifiably countable.	3. It should be difficult to counterfeit, and the genuine must be easily recognisable.
To be anonymous (common to all) :		
<ol style="list-style-type: none"> 1. Money should not be subject to government tracking. 2. It should be useable for purchases in a black market. 3. It should not require equipment, tools or electricity to use. 		
Money also is typically that which has the least declining marginal utility, meaning that as you accumulate more units of it, each unit is worth about the same as the prior units, and not substantially less.		

Table 1. Characteristics of money

reinforcing social ties and location. A definition providing room for sustainable development is needed, eliminating the hierarchical social systems whose formation is encouraged by centrally controlled money, since in most cases this type of money creates disadvantages when social communities need to be built and sustained over time. For this type of money to be sustainable, social participation needs to be committed to developing trust between both individuals and social groups, extending to nations or geographically distributed governments. If the idea of modern mutual banking was proposed only in the 19th century (by French social-anarchist Proudhon [4]), it is possible to see that, compared to human trade, this type of money has existed over a historically short time-span, and changes in attitude to money and shifts in social practices in the near future cannot be ruled out. Such an approach can create a shift in our understanding of money as a meta-model monopolising control over the issue of value and currency.

These ideas are not new; in the early 20th Century Silvio Gesell – the founder of the school of Free Economics – envisaged money liberated from interest, or even carrying a negative interest [8]. The acquisition of interest for lending money is one of the major preoccupations linked to the use of money; in fact, the demand for interest historically strengthened the concentration of money in the hands of the rich as a way of further financing the expansion of national economies, imposing demands on future economic resources. Gesell disputed this idea. In his eyes money was in fact a service of the community to the users [9].

Gesell proposed that everyone having money in their hands at a certain point of time should pay a liquidity tax on the possession of money; in his view this tax would make people want to lose the money

as soon as possible. They would buy something with it, invest it in durable goods or lend it, so that money had its unlimited exchange function again, without the need to offer the money's owner profit on lending. Gesell thought that this practice would liberate the economic system from the grip of interest. In our times with computer technology this idea is feasible. Lietaer, who coordinated from Brussels the introduction of the European Monetary Union (EMU), follows Gesell; in some way he is also seeking to use money without the liability of interest; in other ways, there is a renewed search for a system in which people are stimulated to spend their money directly and at the same time to take care of the future [9].

Lietaer's view reduces the functions of money to two: money as a measuring unit, and money as a unit of exchange that can be abstracted further by trading money via electronic systems. In principle society creates money and transforms it according to their needs. Already in Stock Exchanges all over the world, gold and paper money can be traded electronically via online systems. It is speculated that in the near future developed economies may migrate from the present system of physical money to an entirely digital currency system. Money is becoming a form of social information, hence the need to re-evaluate what money is, and its real contribution to the management of material and human resources available now and in the future.

The vision of Gesell and Lietaer can be easily implemented with modern real-time computer technology as the money functions for storing and exchanging are merged into one. Money circulates faster and faster as electronic systems are in place, hence money owners either circulate it or lend it. Because of this it is possible to stimulate an economy without the involvement of interest. Once interest is not required, the idea of money tenure has changed. The idea of future income will be directed to consuming products with a longer life-span, which might be perceived as reducing the stimulus for economic growth, but from another perspective it can also be seen as encouraging the development of a more durable and stable economy. These types of thoughts are basic assumptions in the models that will be explained in Sections 3 and 4, in which sustainability is at the core of the socio-economic models presented.

For the purposes of this document, the economics of sharing discussed below as one of the phenomena underpinning community currencies is mainly focused on sharing unused capital and does not include the Open Source phenomenon or the processes and interactions collectively referred to as the Gift Economy. It is beyond the scope of this paper to present an in-depth analysis of the many ideas of money. Our emphasis, rather, is to introduce the reader to the basic ideas linked to Community Currencies and Economics of Sharing applicable to mobile technology and its applications.

3 Community Currencies as a Viable Working Framework

In the discussion of how community currencies can be used as a viable working framework for mobile technology applications, we need to understand in the first instance the shift in the ideas about money presented above, requiring stronger community participation. In any case, the view in this document is to propose the use of community currencies as complementary to established economic models. Community participation is seen as expressing real and virtual social networking links. If the use of money or community currencies can be explained in parallel to the historical evolution of the idea of "Commons" as a means for structuring rights to access, use and control over resources [10], then working commons need to define in the first instance their accessibility, either as restricted to members or open to all. Secondly, commons need to define whether their workable system is regulated or unregulated. Depending on which combination of these two parameter definitions is workable, four types of commons can be classified (see Table 2 below).

The idea of the commons can be extended to community currencies as they have traditionally been geographically localised and their economic interconnections tend to be of a social-geographical nature [1]. There have been many local money systems throughout history, which have merely been small-scale versions of the larger national currencies. But these work no better at the local level than they do at the national. Issued in scarce supply by some local or regional authority, such currencies, by their very nature, create a local context of competition, which in turn generates conditions for local unemployment, local rich and local poor. Furthermore, they are inherently even less stable than their national counterparts, and prone to embarrassing and irrecoverable collapse [11].

	Restricted accessibility	Open accessibility	Regulated	Non-Regulated
Open Commons	N.A.	Examples: Oceans, Air. There is no access restriction	Sidewalks, streets, roads and motorways that are not tolled	Open Access: not regulated by any rule
Closed Commons or limited common property	Toll roads, Limited pasture arrangements	N.A.	Most limited commons have some type of regulation	Very rare, few cases

Commonality: The inputs and outputs of the sharing process are shared, freely and without conditions, in an institutional form that leaves all accessing members free to use the resource as they choose.

Table 2. Commons as a unit of cohesion for Community Currencies (Benkler, 2006)

To overcome these limitations, three major issues in the creation of money have to be addressed, whereby the faults in the previous model of money are avoided. Community currencies rely upon addressing such issues [11]:

- Community currency aims at least to provide money that exists in sufficient supply,
- Such money is only for use within a community (it establishes a closed loop)
- Ideally this money will be created by its users (sourced from within the community).

Following the metaphor of the Medieval village discussed in Section 1, this small rural community can be compared to its counterpart in modern times, a country village. Assuming that external factors such as inflation and interest are kept as economic externalities for the value of money as a medium for trade, the phases described are valid. Without the existence of a community currency, the money saved by the village is invested or redistributed in the urban areas. This way the village(s) falls into economic decay. This cycle is illustrated in Figure 2. By contrast, the use of community currencies or a local savings bank can strengthen the local economy as it offers a way of keeping the money in local circulation for longer. This cycle is illustrated in Figure 3.

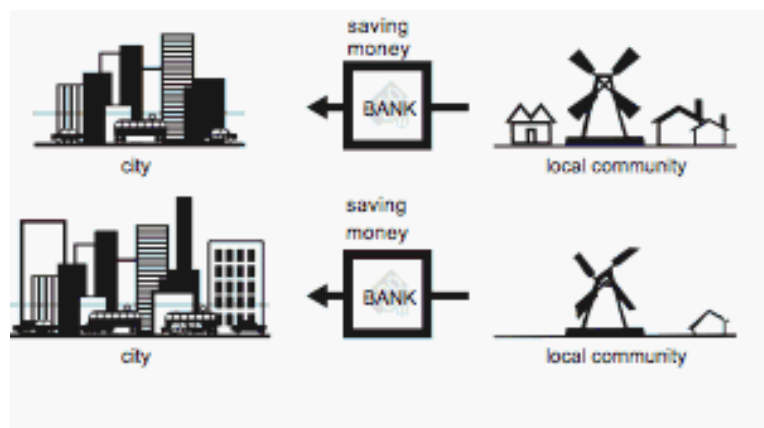


Fig. 2. Saving money from village is invested in the urban areas. As a consequence, village falls into economic decay (Source from Van Arkel and Peterse, 1998)

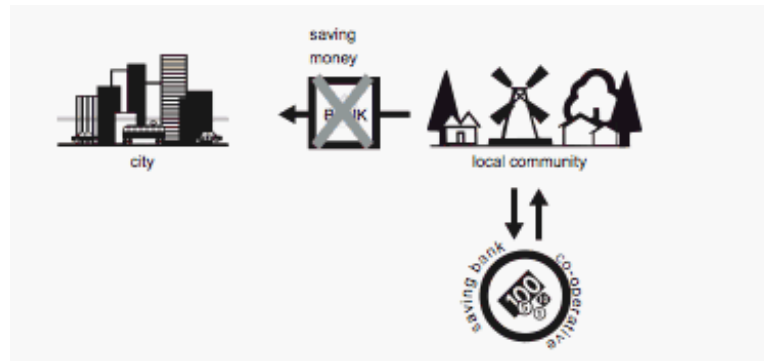


Fig. 3. Using local bank or community currency strengthens the local economy (Source from Van Arkel and Peterse, 1998)

The next question to ask in this socio-economic model is where and for what purpose is the money circulating in the local community used? The simplified idea of this model is that consumers (the villagers) use the payment from their wages to pay for purchases from the producers in a permanent trading loop. For a small community, such as the country village described above, it is a certain reality that most of the money gained by the villagers in wages is used for the exchanges of goods and services outside the local community. These outside payments create a disruption in the circulation of money, in which as purchases are processed there is a leak in money for wages, which in turn leads to smaller/fewer purchases within the community and bigger leaks in money outside the community.

The money leak goes to established channels of capital: either for interest payment, profits, compensation for land, raw materials, private savings, payments for patents, etc. The financiers or managers of this capital might be placed in a situation in which they receive more money than they can invest, which can lead to a decrease in wages and purchases, creating an economic crisis. Money can also leak through loans of the bank to be used by the local community, which is then obliged to make interest payments to honour the debt. The analogy of a balloon economy can then be used: money is issued or created by general or national banks to compensate for the money that disappears in the balloon of capital, making the finance sector extremely powerful in dictating economic policies. The money therefore fails to perform its function as a medium. At the same time governments can borrow extra money and spend it and stimulate purchasing power this way. However the interest payments to be paid for such loans will create yet another leak in the monetary system.

The development and use of community currencies is a way of reducing the influence on such circles of financial sourcing: by keeping resources local there is a significant amount of re-use and reparation. Whilst this does not lead to growth in the local economy, it does limit the dependency on external resources. There are combinations of community currencies, such as Noppes [9], which uses LETS, focused on private individuals within a barter system. There are limitations in the way community currencies have been able to bring business, public and social services together in systems such as Ithaca Hours and LETS [12]. These models exclude members that cannot offer goods or services for trade, modelling the community currency on the business-to-business nature of commercial barter associations, by which LETS is loosely inspired. *LETS are schemes to encourage people to exchange goods and services within their local communities, creating an alternative economy outside and parallel to the wider money economy as a systematic form of barter* [2].

What LETS and other similar community currency trials have done is to help rebuild localised economies by making them less reliant on outside sources of goods and services, as well as promoting a sense of community between members. This is a departure from the traditional economic model in which such sense of community has no economic value whatsoever. Community currency frameworks can potentially succeed in reviving a depressed local economy. This can be achieved by facilitating money exchange in situations in which money itself achieves a sub-optimal level of distribution. Community currencies have the ability to increase the local levels of exchange, but they are in some ways seen as a way of promoting social justice by redistribution through mutualism. Most community currency systems

operate thanks to the tireless efforts of volunteers working with shoestring budgets, creating burdens that are difficult to reconcile with long-term views of using such currencies.

In the case of the country village metaphor used to illustrate the economic principles in which community currencies are sustained, there are without doubt direct benefits in adopting community currencies in parallel to the ordinary currency. This view is not normative but auxiliary in helping with the local community trading issues. What we are seeing now is the emergence of more effective collective action practices that are decentralised but do not rely on either the price system or management structures [10]. Community currencies fall into this category since networked environments provide a more effective platform for the actions of non-profit organisations spreading their activities over a wider community.

[13] proposed that what is being seen around the world is the creation of new transaction media, largely due to technology and in particular the medium of the Internet; examples include e-cash, e-gold, etc. This trend leads to trade utilising multiple parallel currencies: local, national and international. There is no need to choose one currency over the other as they could all coexist, closing a circle of economic models that have been tried since Medieval times. It seems that in order to establish trust among the community members, the formation of closed commons in which there are regulated norms for trading and membership is more successful than open commons in which anyone can participate. This seems to be the case for most online communities that have embarked on these types of schemes. The discussion then has to reflect upon the motivations for trading or sharing that lead to the search for such economic models, as will be presented in the section below.

4 Economics of sharing

Community currencies are most effective when there is a culture of shared ownership or shared distribution in the social interactions in which money is a medium for trade. Most of the contemporary economic models adopt a very simple model for human motivation: the basic assumption is that all human motivations can be more or less reduced to positive or negative utilities: things people want, and things people want to avoid. These utilities are all represented by the money or currency exchange by the society members. There is plenty of room to discuss these assumptions, which are obviously not always correct in representing the diverse social frameworks in which humans operate. Community currencies' continuous presence in social history has shown that humans do have other motivations besides the accumulation of money for their consumption of utilities, especially if there is a long-term view of how to use and manage human and natural resources.

Like any other economic theory dominant in the 20th Century, the utility assumption is currently subject to a revisionist process, where economic models can widen their assumptions for human motivation by creating a distinction between intrinsic and extrinsic motivations. Extrinsic motivations are imposed on humans by external factors, and social pressure then reduces the room for intrinsic motivations. There are two rewards associated with human beings in a social structure: economic and social standing [14]. As technological developments make society better networked, new ways of social production emerge [15]. Even so, *post-modern sharing is emerging from certain physical, rivalrous goods and will increase due to advances in technology* [10]. Until now this sharing has not been extended to the spheres of *social sharing, in which a third mode of organising economic production, alongside markets and the state* [16], is being built up.

The quantification of social sharing in monetary terms is linked to intrinsic motivations such as the need for belonging to a community and being valued not only by the products or goods consumed, but also through the exchange of money, which better expresses this post-modern sharing of resources. The economics of sharing is departing from these models of human motivation, searching for a motivation for human economic activity that goes beyond the value of human actions based on money exchanges, building upon human capital to create a more optimal distribution of wealth and resources that give not only a material but also a spiritual reward (e.g. a better environment, social recognition, reinforcement of community ties). What is understood as sharing is the ability to motivate human interaction or sociality besides the need to be awarded money for exchanging goods and services. The source of this type of sharing is a way of providing an interpretive and subjective space in which humans can allocate a subjective value to the sharing.

BIONETS is a technology platform in which the opportunity for the creation of virtual spaces with a subjective value to the sharing can co-exist with known economic models provided by the telecommunications industry. Its case is not unique; current economics of sharing models contextually attached to the use of mobile technology are evolving over service platforms that are social networks in digital forms; the driver to share is expressed in several modes, such as:

- Sharing and distribution of electronic tokens between social networks of friends using I-mode enabled phones in Japan [17]
- Sharing tokens, or paying for services through mobile services that are centralised, such as car parks, vending machines, and others.
- Sharing contribution to the fitness valuation of mobile services, sharing moods or opinions about service quality.
- Bluetooth/Wifi sharing of information in digital format (MP3s)
- Video Sharing (www.mytube.com), now in mobile format

These are basic forms of sharing that have been adopted by many mobile users around the world; a valid question will be to ask how a sustainable economic model of sharing, developed through mobile services, can be defined. Until now, sharing has provided only limited exchange between small groups of friends (e.g. Java applets) but has not linked such exchange to any monetary value. It is possible however to forecast a near future in which the economics of sharing will develop to produce exchanges in which the interest shown by the community of users would account for a variable exchange value for the goods or services exchanged.

5 Technology as a mediator in the use of money and Empirical Business Models: Working Cases

Following the arguments presented in Sections 1 and 3, technology and in particular the use of the Internet raised the possibilities and expectations of using technology as a replacement for physical money, e.g. the increased use of electronic money by Stock Exchanges. This is a relatively new idea, both for the financial world and for ordinary people. However, sharing is common in the computing world at all levels: primary computing power (e.g. computer grids), code development using models such as Open Source, and information services such as blogs, wikis and newsfeeds. When the sharing model has been extended by the mediation of technology in other economic areas, for example the case of the users of LETS, it has been found that – assuming in the case of LETS a strong bias towards making the model succeed for ideological rather than economic motivations – technology is a way of reducing the ideological element, increasing participation, simplifying adoption and making the model become truly economically motivated. Users and the communities they belong to can easily use the technology to share resources and exchange virtual forms of currency with a subjective value. Examples of working community currencies that are mediated by technology include:

LETSystems: <http://www.gmlets.u-net.com>

This model is probably the best known and most popular among all the already established community currencies. LETS was intended to build local currencies into a sustainable social system according to five criteria [2]:

- It is non-profit making
- There is no compulsion to trade
- Information about balances is available to all members
- The LETS unit of credit is equal in value to the national currency
- No interest is charged or paid.

The participants in a LETS scheme can trade goods and services with each other using a group's own local currency, the value of which is usually matched to an exchange rate in parity with the national currency. Members are encouraged to trade a wide range of goods and services, including those that

often might appear as redundant in modern society. Members search the electronic catalogue until they find goods or services they require, telephone or trade online with the person or group trading those goods or services, and negotiate a price in the LETS local unit of credit. The buyer then issues an order for purchasing the goods that is sent electronically to the LETS system. Their account is debited with the amount and the seller's account is credited with the amount in the order. Unlike national currency, LETS credits do not have to be earned before they are spent. Local currency is not issued at national but at local level. Therefore, this money is theoretically unlimited and economic activity is not restricted or constrained by lack of money or money leaks in the local economy. No interest is payable on either negative balance or savings. If a member has a negative account balance, this balance is interpreted as a commitment of the member to supply goods or services in future. There is no incentive for people to accumulate savings, because the money is worthless outside the group. The quick circulation of the local currency by earning and spending can and should be encouraged.

Ithaca Hours: <http://www.ithacahours.org>

The Ithaca Hours is a local currency system that promotes local economic strength and community self-reliance in ways which support economic and social justice, ecology, community participation and human aspirations in and around Ithaca, New York. Ithaca Hours helps to keep money local, building the Ithaca economy. It also builds community pride and connections. Over 900 participants publicly accept Ithaca Hours for goods and services. Additionally, some local employers and employees have agreed to pay or receive partial wages in Ithaca Hours, further continuing the goal of keeping money local. This model is a commons restricted to the citizens of Ithaca.

Salt Spring Island Dollars: <http://saltspring.gulfislands.com/money/welcome.htm>

The citizens of Salt Spring Island looked at the LETS and Ithaca Hours models and identified two problems with the use of such community currency models: one problem was the reluctance of the majority of merchants in this area to accept such currency unless the currency was 100 per cent redeemable into their national currency; the second problem was how, when the currency was in circulation, to receive and/or maintain any value for the holder. In addition, the operation of local currencies was segregated from local financial institutions. The way these problems were addressed was to give the currency a two-year expiry date; any bills not redeemed by the expiry date would represent a profit. By creating a profit concept, to cover costs, it made it possible for the currency to go into circulation through a one-to-one exchange with the Canadian Dollar. The system went live in 2001. Given that the island has a population of around 10,000 with an annual tourist flow of over 200,000, a potential edition of 20,000 has a considerable base. The currency is now issuing coins made of precious metals and complies with all the aims and goals that other community currencies have established.

Strohalm: <http://www.strohalm.org/vlcs.html>

The Social Trade Organisation, STRO (in Dutch STROhalm) is a Research and Development network working in the field of micro-credit, development of small and medium enterprises and strengthening of local economies through complementary circuits of exchange. The basic tool of STRO's methodologies is a circuit of exchange between producers and consumers. The part of the flow of money that enters a community through the purchasing power of these consumers, or connected governmental agencies, is circulating several times through the exchange circuit before it leaves the area, to bring as much unused capacities as possible back into productivity. Some of their projects aim to lead on the very first step towards entrepreneurship, whilst others are more elaborated tools to optimise the outreach of the existing economic actors. All projects aim to result in more employment, income and opportunities for local communities to improve their living conditions. Most of the pilot projects are concentrated in Brazil, Central-America, Asia and the Netherlands.

Brazilian Educational Currency (Saber) [18].

The main learning focus of this currency is to multiply the number of students that can afford to obtain a college-level education in Brazil. The approach is to create a special targeted currency, whose unit

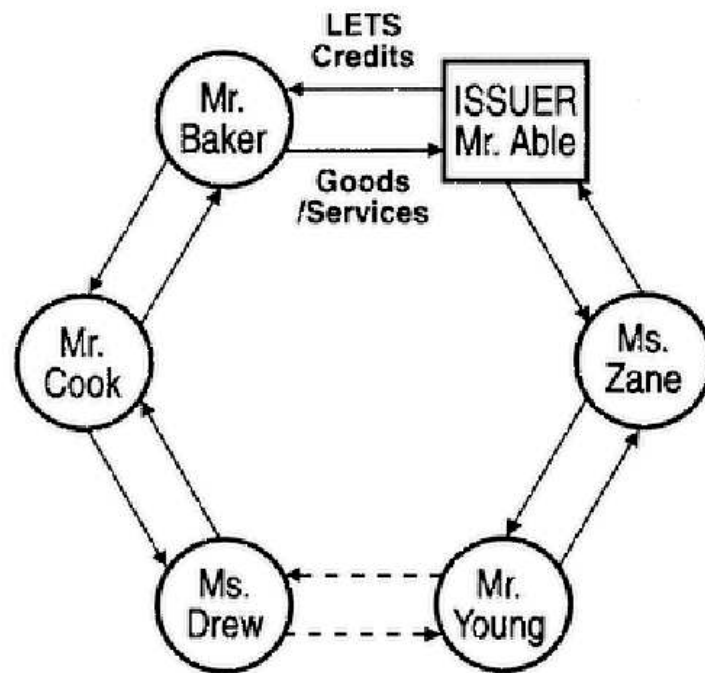


Fig. 4. An illustrative model of how the LETS credits can be used within a local community: A self-sustaining mutual credit trading cycle (based on www.ratical.org (2008))

is called Saber, which would be issued under highly controlled conditions in the educational system. Its face value would be nominally the same as a Real (Brazilian currency), and would be redeemable for tuition in higher education programs in participating universities. This would be a paper currency (although electronic accounts can be kept where they will accumulate), with all the security precautions against fraud used for printing conventional national currency. An illustration of how this currency is being implemented in Brazil is shown in Figure 5. The process includes three stages. Students that participate in this community currency from an early age will reach the university level with considerable redeemable currency to exchange for real money. The currency is still in its early stages and there is no data to confirm its success or the model’s validity.

Business models already in place using mobile technology that enable community currencies, and setting up processes aimed at furthering the economics of sharing, are listed below:

Exchange through mobile phones: http://www.economist.com/displayStory.cfm?story_id=8089667

Money transfer between rural and urban areas. Most poor people in South Africa do not have bank accounts, yet most own or have access to a mobile phone. This economic transfer model is completed by using mobile phone text messages as a way of certifying the deposit of amounts of money to a trader, who then verifies the exchange with other traders in other places, paying for their goods by credit based on the text validation.

Starbucks coffee card: <https://www.starbucks.com/card/default.asp?cookie%5Ftest=1>

This is an example of a “commercial” community currency aimed to be used as a local, national and international medium of exchange. This is an example of an international company creating a community currency for coffee consumers. The idea is that clients exchange their money for credit on a card that can then be used in any other local or international Starbucks franchise to purchase goods. It simplifies the need for small change to pay for coffee, and can be used anonymously, whilst still allowing Starbucks to keep track of purchases.

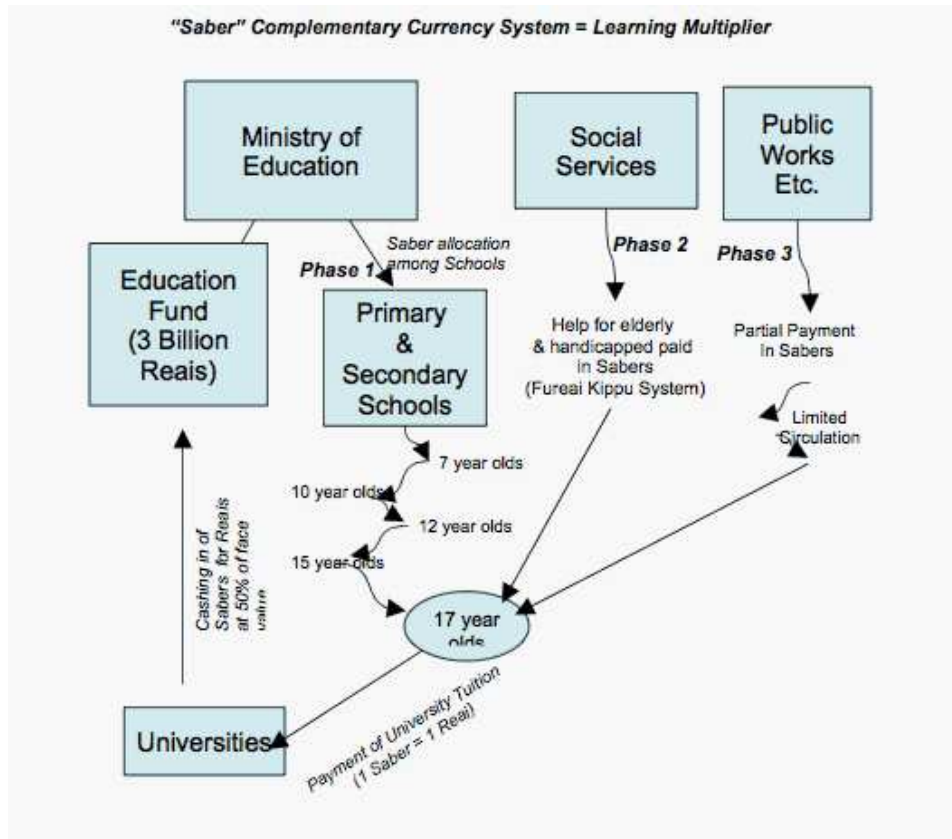


Fig. 5. Saber Complementary Currency Model (from [18])

The SIMS: Simelons currency: <http://thesims2.co.uk>

This is an example of a virtual currency with no defined exchange rate with real currencies (you need to be logged in) linked to the SIMS players community. However the success of the SIMS has created a trading exchange of SIMS code in the auction market. On Ebay, for example, it is possible to buy houses designed for the SIMS that can be highly valued and once purchased can be uploaded in the SIMS game and integrated into the game.

LindeX Second Life Virtual World Currency: <http://secondlife.com/whatis/currency.php>

This community currency is used by the members of Second Life. The conversion rate between real currency and LindeX is currently 1 USD = 300 LindeX. The currency has its own stock market, and it is used to purchase all types of goods and services in the virtual community; it is a closed common, and extremely popular.

My gamma: <http://en.wikipedia.org/wiki/MyGamma>

This community currency does not exchange money as such but knowledge, social networks, friendships and other tokens. It is particularly popular in Asia. The trading is also token-based and the value assigned to some of those tokens is subjective and depends upon the demand for them. These models are based on sharing common resources provided by technological development based on the theoretical socio-economic models; there are however additional characteristics of note:

- The sharing is free from geographical restrictions.
- When there is a virtual or real currency to be traded, the value assigned to the unit is universally accepted by all partners.
- There is a conversion exchange rate between the virtual and the real world.

In real terms the mediation of technology allows the easy creation, accountability, and trade of virtual currencies, some of which follow the conventional idea of money discussed in Section 1 of this paper,

whilst others have followed the path established by community currencies. However it is clear that this trend in associating parallel currencies with national currencies will continue and become more sophisticated as technology advances further, perhaps making incursions into areas of trade currently based on conventional models. Nascent technologies which might further this trend include exchanges of identity and DNA samples. It is important to understand that the creation and measurement of wealth will have to be reconceptualised to include the new exchange opportunities opened to all people through the technology. In the end, alternative business models are a clear showcase for a basic economic principle of the Internet: that even though money doesn't change hands, attribution is a valuable economic right in the information economy [19].

6 Further research applied to BIONETS: Developing a Sustainability Model

This paper has presented an overview of how money as a concept has changed over time as well as a compendium of knowledge about social economic models for Economics of Sharing and Community Currencies from a social ontological framework. The aims of this paper are limited to a sketch of the economic terms currently in use by both the economics of sharing and community currencies communities, presenting the case for its use for money as a medium for digital transactions. The relevance of such concepts applied to the BIONETS service creation development model can be seen as self-evolving, autonomic, aimed at gaining the best support for user tasks while reducing the user's efforts in service creation. From this point of view, there exist opportunities to research scenarios where the technology is used selectively, both by individuals and by the social networks linked by the distributed applications available from BIONETS services. There are already indications that such platforms will lead to an exchange of services and goods beyond those based on the technology.

For example, researchers at Nokia speculate that, within a decade, the cost of storage will have fallen so far that it might be possible to store every piece of music ever recorded on a single chip that could be included in each phone [20]. It would be necessary to update the chip every so often to allow for new releases, of course. But this could open up new business models that do not depend on downloading music over the airwaves; instead, the phone could simply exchange brief messages with a central server to unlock purchased tracks or report back on what the user had listened to for billing purposes [21]. This type of exchange will raise the opportunity to create communities in which music recording might be the currency of trade associated with that community.

There is also the possibility of using mobile phone applications to confirm transactions with one another by simply pressing a button. A central or distributed server could confirm that a transaction has been completed. This will be similar to enabling the back-end for a truly decentralized marketplace with buyers, sellers, traders, and sharers "in an Open Source killer app for the 21st Century" [22]. The future possibilities for alternative economic trading are already in place, well beyond the original conceptual design from engineers and business analysts. The users and their communities are constantly evolving and developing new ways to use the technology for their own purposes. This is the medieval metaphor emerging again in our times aided by technology. The physical limits for trading of the medieval village are now eliminated by creating a global village that is both virtual and regulated by its users.

In terms of the BIONETS service layer, as the project advances and applications are developed using the BIONETS infrastructure, it will be possible to put to test some of the concepts discussed in this paper. Special mention can be given to the designed *undersound* [23] application where there is the opportunity to create a community currency trade token based on music. Another possible environment for the use of such tokens can be the BIONETS Personal Platform, currently under development by the project partners. However a major concern for the development of such applications is the sustainability model that makes such applications successful over time. One major concern is the changes to the platform (e.g. migration of games valid on one type of mobile phone to another), and keeping profiles and communities that can be transient or highly dynamic over time during their exchanges through the media.

As an introduction to the concepts in the area of money, community currencies, and economics of sharing, this paper has aimed to link the ideas to the applications and services already available, as well as to explore some the benefits and drawbacks of such uses.

References

1. J. Schraven, "The economics of community currencies: a theoretical perspective," 2001, unpublished Honours Thesis, Oxford University, [http : //www.jorim.nl/economicscommunitycurrencies.html](http://www.jorim.nl/economicscommunitycurrencies.html).
2. C. Cadwell, "Why do people join local exchange trading systems?" *International Journey of Community Currency Research*, vol. 4, 2000.
3. "The economics of sharing," 2005, [http : //www.economist.com/finance/displayStory.cfm?storyid=3623762](http://www.economist.com/finance/displayStory.cfm?storyid=3623762).
4. Wikipedia, 2006, available at www.wikipedia.org, online source.
5. K. Menger, "On the origin of money," *The Economic Journal*, vol. 2, pp. 239–255, 1982.
6. Reisman, *Capitalism: A Treatise on Economics*. Ottawa, Canada: Jameson Books, 1990.
7. F. Cesarano, *Monetary Theory and Bretton Woods: The Construction of an International Monetary Order*. Cambridge, UK: Cambridge University Press, 2006.
8. S. Gesell, *The Natural Economic Order*. Berlin: Neo-Verlag, 1929.
9. H. Van Arkel and G. Peterse, *Money and its alternatives: How money controls the world and the alternatives to change it*. Utrecht, The Netherlands: Stichting Aktie Strohalp, 1998.
10. Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven and London, Yale University Press, 2006.
11. M. Linton and E. Yacub, "Open money," 2006, [http : //subsol.c3.hu/subsol2/contributors0/lintontext.html](http://subsol.c3.hu/subsol2/contributors0/lintontext.html).
12. T. Cohen-Mitchell, "Community currencies at a crossroads: New ways forward," *New Village Journal*, vol. 2, 2000, [http : //www.ratical.org/manyworlds/cc/cc@Xroads.html](http://www.ratical.org/manyworlds/cc/cc@Xroads.html).
13. B. A. Lietaer, *The Future of Money: Creating New Wealth, Work and a Wiser World*. Century, 2001.
14. M. Granovetter, "A theoretical agenda for economic sociology," 2000, working Paper.
15. M. Castells, *The Rise of the network society*. Oxford, UK: Blackwell, 1996.
16. Y. Benkler, "Sharing nicely: On shareable goods and the emergence of sharing as a modality of economic production," *Yale Law Journal*, 2004, [http : //www.yalelawjournal.org/pdf/114-2/Benkler_FINAL_LJ114-2.pdf](http://www.yalelawjournal.org/pdf/114-2/Benkler_FINAL_LJ114-2.pdf).
17. H. Rheingold, *Smart Mobs*. Cambridge, USA: Perseus Publishing, 2002.
18. B. A. Lietaer, "A proposal for a brazilian education complementary currency," *International Journey of Community Currency Research*, vol. 10, pp. 18–23, 2006.
19. M. Shiels, "Legal milestone for open source," 2008, [http : //news.bbc.co.uk/1/hi/technology/7561943.stm](http://news.bbc.co.uk/1/hi/technology/7561943.stm). (Last visited August 2008).
20. G. T. F. E. I. Unit, "The phone of the future," 2006, [http : //globaltechforum.eiu.com/index.asp?layout=rich_torychannelid=3&categoryid=1&title=The+phone+of+the+future&docid=9762](http://globaltechforum.eiu.com/index.asp?layout=rich_torychannelid=3&categoryid=1&title=The+phone+of+the+future&docid=9762).
21. "The phone of the future," 2006, [http : //www.economist.com/printedition/displayStory.cfm?storyid=8312260&fsrc=RSS](http://www.economist.com/printedition/displayStory.cfm?storyid=8312260&fsrc=RSS).
22. D. Rushkoff, "Open source currency," *The Feature*, 2004, [http : //www.rushkoff.com/TheFeatureArchive/oscurrency.html](http://www.rushkoff.com/TheFeatureArchive/oscurrency.html).
23. A. Bassoli, "undersound: exploring the experience of riding the underground," *Media and Communications Department, The London School of Economics and Political Science*, vol. 37, 2006.

Author Index

Al Hanbali, Ahmad 177
Altman, Eitan 35, 54, 225
Avrachenkov, Konstantin 260

Battiti, Roberto 260
Bernhard, Pierre 225
Brunato, Mauro 7, 260

Carreras, Iacopo 177
Cascella, Roberto G. 260

Debbah, Merouane 225
Dini, Paolo 67, 105

El-Azouzi, Rachid 35
Elaluf-Calderwood, Silvia 279

Fiems, Dieter 54
Florent Benaych-Georges 201

Gomez, Karina M. 44

Hayel, Yezekael 35
Horváth, Gábor 105

Ibrahim, Mouhamad 177

Mähönen, Petri 241
Merouane Debbah 201
Miorandi, Daniele 23, 44

Neglia, Giovanni 251
Nemirovsky, Danil 260

Oldewurtel, Frank 241

Pham, Son Kim 260

Riihijäervi, Janne 241

Schreckling, Daniel 67, 105
Silva, Alonso 225
Simon, Vilmos 177

Tembine, Hamidou 35

Varga, Endre 177

Yamamoto, Lidia 23, 44

Subject Index

- Activator–Inhibitor 44
- Activator-Substrate Model 49
- Ad Hoc Networks 177, 251
 - Optics and Electrostatics Models, 225
 - Road traffic Engineering, 225
- Algebra
 - abstract, 67, 70
 - Boolean, 92
 - coalgebra, 157
 - cosets, 72
 - factor rings, 74
 - field extension, 76
 - fields, 71, 75
 - groups, 71
 - homomorphisms, 72, 74
 - ideals, 74
 - image, 74
 - kernel, 74
 - quantifier, 95
 - rings, 71
- Algebraic Automata Theory 105
- Algorithmic Chemistry 29
- Algorithmic game theory 252
- Allocation Rule 254
- Amorphous Computing 45
- Artificial Embryogeny 23, 31
- Artificial Evolution 24
- Artificial Regulatory Networks 28
- Autocatalysis 46
- Autoconstructive Evolution 27
- Automata
 - Specification, 147, 150
 - Theory, 147
- Automata Specification 150
- Automata Theory 147
- Biology
 - Cell, 67, 105
- Branching processes 54
- Brown – von Neumann – Nash dynamics 39
- Brussellator 47
- Catalan Numbers 209
- Category Theory 105, 142, 151, 153, 157
 - Adjunction, 160, 167, 169
 - Behaviour, 164
 - counit, 170
 - specification, 164
 - unit, 170
- Cell Biology 67, 105
- Cell Differentiation 45
- Cellular Automata 50
- Cellular Neural Networks 50
- Chemical Computing 29
- Chemical Genetic Programming 28
- Coalgebra 157
- Code
 - self-modifying, 27
 - self-replicating, 27
 - self-reproducing, 27
- Computational Evolution 24
- Congestion Control 40, 51
- Connection Games
 - global, 257
 - local, 255
- Delay Tolerant Networks 64, 177, 186
- Distributed Coordination 44
- DTN *see* Delay Tolerant Networks
- EDA *see* Estimation of Distribution
- Embryogeny
 - artificial, 23, 31
- Embryonics 30
- Epidemic Routing 177
- Epigenetic Programming 28
- ESS *see* Evolutionary Stable Strategies, 36
- Estimation of Distribution 15
- Evolution 107
 - Computational, 24
- Evolutionary Algorithms 24
- Evolutionary Computation 24
- Evolutionary Computing 23, 24
 - Dynamic Optimization Problems, 26
 - indirect encoding, 28
- Evolutionary Games 35
 - Evolutionary Stable Strategies, 35
 - Replicator Dynamics, 38
- Evolutionary Stable Strategies 35, 36
- Ferry–based Wireless Local Area Network 56
- Fluid Models 225
- Free Convolution 202, 212, 213
- Free Probability Theory 201
- Functional completeness 109
- Game

- evolutionary, 35
- global connection, 257
- Local connection, 255
- Network Formation, 253
- Game Theory 35, 252
 - Algorithmic game theory, 252
 - Allocation rule, 254
 - Hawk and Dove, 37
 - local connection, 255
 - Nash Equilibrium, 253
 - Network Design Games, 251
 - Network Formation Games, 251
 - Pairwise stability, 254
 - Price of anarchy, 255
 - Price of stability, 255
 - Value function, 254
- Gene Expression 28, 113
- Gene Expression Programming 29
- Genetic Algorithm 12, 18, 25
- Genetic Programming 25
 - bloat, 26
 - intron, 26
 - parsimony, 26
- Ggenetic Algorithm 12
- Gierer-Meinhardt model 47
- Gossiping 181
- Gossiping search heuristics 19
- Graph Centrality Measures 263
- Groups 126
 - Cayley’s theorem, 131
 - free, 168
 - Lie, 109
 - semi, 140
 - symmetry, 128, 131
- Hawk and Dove Game 37
- Infinite Server Queue 61
- Intelligent Optimization 7
- Interaction Computing 105, 113
- Kriging 14
- Learning
 - Machine, 7
- Lie groups 109
- Logic 67, 150
 - algebraic, 92
 - branching time, 100
 - first-order, 88, 90, 95
 - linear time, 98
 - propositional, 88, 92
 - temporal, 98
- Logistic Equation 39
- Machine Learning 7
- Marchenko–Pastur Law 207
- Maximum Clique Problem 18
- Medium Access Control 50
- Memetic Algorithms 13
- MIMIC *see* Mutual-Information-Maximizing Input Clustering
- Money 279
 - Community Currencies, 283
 - Economics of Sharing, 286
 - Historical Background, 280
 - Scale-Dependent Phenomenon, 279
- Morphogenesis 44
- Multiple Access Protocols 39
- Mutual-Information-Maximizing Input Clustering 15
- Nash Equilibrium 35, 253
- Network Coding 79
- Network Design Games 251
- Network Formation Games 251
- Networks
 - Ad Hoc, 177, 225
 - Delay Tolerant, 177
 - Dense, 226
 - Formation Games, 253
 - Mobile, 177
 - Pairwise stable, 254
 - Peer-to-Peer, 63, 251
 - Scale-free, 241
 - Small-world, 241
 - Structure, 241
 - Topology, 241
- Newton-Girard Formula 205
- Ontogenesis 24
- Ontogenetic Programming 28
- Ontogeny 24
- Opportunistic Communications 177
- Optimization 7
 - Dynamic, 26
 - evolutionary computing, 26
 - genetic algorithm, 18
 - genetic algorithms, 12
 - gossiping, 19
 - Intelligent, 7
 - large-scale structure, 9
 - meta-heuristics, 10
 - model-based, 14
 - neighbourhood structure, 8
 - population-based algorithm, 18
 - reactive search, 16, 18
 - simulated annealing, 11, 17

- tabu search, 11, 18
- variable neighbourhood search, 17
- Organisation 112
- PageRank 262
 - distributed, 272
- Pairwise Stability 254
- Pattern Formation 44
- PBIL *see* Population-Based Incremental Learning
- Phylogenesis 24
- Phylogeny 24
- Population-Based Incremental Learning 15
- Power Laws 241
- Preferential Attachment 243
- Push (programming language) 27
- Random Graphs 241
- Random Matrix Theory 201
- Reaction–Diffusion 44
- Reactive Search 16
- Reactive search 18
- Replicator Dynamics 38
- Reputation 261
 - Aggregation, 266
 - BlockRank, 270
 - Decomposition, 266
 - Distributed PageRank, 272
 - Eigenvector, 260
 - Fast Two-Stage Algorithm, 271
 - Full aggregation, 267
 - Graph based, 262
 - Measure, 260
 - Partial aggregation, 269
- Response Surface 14
- Routing 51, 227
 - Epidemic, 177, 190
- Scale-Dependent Phenomenon 279
- Scale-free networks 241
- Security 67, 87, 105
 - Reputation, 260
 - Symbiotic, 105, 151
- Self-modifying code 27
- Self-replicating code 27
- Self-reproducing code 27
- Semi-linear Process 57
- Semicircle Law 207
- Simulated Annealing 11, 17
- Specification
 - behaviour-based, 111
- Stochastic Differential Equations 59
- Stochastic Search 7
- Symbiotic Security 105, 151
- Symmetry 108, 110, 128, 131, 134
- Tabu Search 11, 18
- TCP 40
 - AIMD, 40
 - Delay, 40
 - Stability, 40
- Time Delays 38
- Transport Protocols 40
- Trust 261
- Value Function 254
- Variable Neighbourhood Search 10
 - variants, 10
- VNS *see* Variable Neighbourhood Search
- Wardrop Equilibrium 236
- Wireless Sensor Networks 251
 - Optimal Deployment, 228
- Wishart Matrices 212



Paradigms for Biologically-Inspired Autonomic Networks and Services

The BIONETS Project eBook

Edited by:
Eitan Altman
Paolo Dini
Daniele Miorandi
Daniel Schreckling

This work has been partially supported by the European Commission within the framework of the BIONETS project EU-IST-FET-SAC-FP6-027748, www.bionets.eu. This eBook constitutes project deliverable D0.2.3, and its diffusion level is marked as Public.

© BIONETS Consortium, 2010

