D2.1.1 Paradigms and Foundations of BIONETS research
<table>
<thead>
<tr>
<th>Reference:</th>
<th>BIONETS//wp2.1/0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category:</td>
<td>Deliverable</td>
</tr>
<tr>
<td>Editor:</td>
<td>Eitan Altman (INRIA), Paolo Dini (LSE), Daniele Miorandi (CN) Daniel Schreckling (UNIHH)</td>
</tr>
<tr>
<td>Authors:</td>
<td>Eitan ALTMAN, Konstantin AVRACHENKOV, Roberto BATTITI, Mauro BRUNATO, Roberto G. CASCELLA, Merouane DEBBAH, Paolo DINI, Silvia ELALUF-CALDERWOOD, Rachid EL-AZOUZI, Yezekael HAYEL, P. MAEHOENEN, Daniele MIORANDI, Daniel NE-MIROVSKY, Frank OLDEWURTEL, Kim Son PHAM, J. RIIHIJAERVI, Alonso SILVA, Daniel SCHRECKLING, Hamidou TEMBINE, Simon VILMOS, Lidia YAMAMOTO</td>
</tr>
<tr>
<td>Verification:</td>
<td>Each Chapter went to one or two rounds of reviews, each round by two referees. The deliverable has not been accepted and went to another thorough process of anonymous review, where each chapter had two referees. Chapters that did not meet the quality standard requested have not been included in the deliverable.</td>
</tr>
<tr>
<td>Date:</td>
<td>November 5, 2008</td>
</tr>
<tr>
<td>Status:</td>
<td>Draft</td>
</tr>
<tr>
<td>Availability:</td>
<td>Public</td>
</tr>
</tbody>
</table>
SUMMARY

This document is the fruit of the collection of paradigms and foundations that had been identified as being relevant to the design, analysis and optimization of autonomous wireless networks, in general, and to the Bionets architecture and services, in particular. Although this paradigm collection is the deliverable of SP2.1, it contains also contribution from other workpackages as well as Chapters that are the result of integration between workpackages. Each chapter in this document contains a survey on a given paradigm. We further included in the chapters specific contribution of the Bionets project to the foundations and paradigms surveyed. The document contains three parts: paradigms from biology, from physics and from social sciences. A preface explains the relations between these areas and Bionet.
# Contents

1 Introduction 13
   1.1 Biologically inspired paradigms. ................................................................. 13
   1.2 Paradigms Related to Physics. ......................................................................... 14
   1.3 Paradigms from Social Sciences. ..................................................................... 15

I Biological Inspired Paradigms 19

2 Machine Learning for Intelligent Optimization
   M. Brunato 21
   2.1 Introduction ........................................................................................................... 21
   2.2 Optimization problems ......................................................................................... 22
      2.2.1 Neighborhood structure ................................................................................ 23
      2.2.2 Large-scale structure ..................................................................................... 24
   2.3 Basic optimization meta-heuristics ..................................................................... 25
      2.3.1 Variable Neighborhood Search ...................................................................... 25
      2.3.2 Simulated annealing ....................................................................................... 26
      2.3.3 Prohibition-based (Tabu) Search .................................................................. 26
      2.3.4 Genetic Algorithms ......................................................................................... 27
      2.3.5 Model-based heuristics ................................................................................... 29
   2.4 Reactive search concepts ...................................................................................... 31
      2.4.1 Variable Neighborhood Search ...................................................................... 33
      2.4.2 Simulated annealing ....................................................................................... 33
      2.4.3 Tabu Search .................................................................................................... 34
      2.4.4 Genetic and Population-based algorithms ....................................................... 34
   2.5 Biology-inspired applications ............................................................................. 34
      2.5.1 A population-based approach to reactive search ............................................ 35
      2.5.2 Gossiping Search Heuristics .......................................................................... 35
   2.6 Relevance to the BIONETS context ................................................................... 37
   2.7 Conclusion ............................................................................................................ 37

3 Evolutionary Games
   H. Tembine, E. Altman, Y. Hayel, R. El-Azouzi 45
   3.1 Introduction ............................................................................................................ 45
   3.2 The framework ....................................................................................................... 46
   3.3 Evolutionary Stable Strategies ............................................................................. 47
   3.4 The Hawk and Dove (HD) Game ......................................................................... 47
   3.5 Evolution: Replicator Dynamics .......................................................................... 49
   3.6 Replicator dynamics with delay ........................................................................... 49
   3.7 Other evolutionary models .................................................................................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8</td>
<td>Application to Multiple Access Protocols</td>
<td>50</td>
</tr>
<tr>
<td>3.8.1</td>
<td>The Model</td>
<td>50</td>
</tr>
<tr>
<td>3.8.2</td>
<td>Delay impact on the stability</td>
<td>51</td>
</tr>
<tr>
<td>3.9</td>
<td>Application to transport protocols</td>
<td>52</td>
</tr>
<tr>
<td>3.10</td>
<td>Conclusions</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td>On Abstract Algebra and Logic: Towards their Application to Cell Biology and Security</td>
<td>59</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Abstract Algebra</td>
<td>63</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Starting Concepts</td>
<td>63</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Groups, Rings and Fields</td>
<td>64</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Cosets and Homomorphisms</td>
<td>65</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Kernel, Image, Ideals and Factor Rings</td>
<td>68</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Fields</td>
<td>69</td>
</tr>
<tr>
<td>4.2.6</td>
<td>Field Extensions</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Logic, Algebra, and Security</td>
<td>74</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Propositional and First-Order Logic</td>
<td>74</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Algebraic Logic</td>
<td>79</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Temporal Logic</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>Conclusion</td>
<td>90</td>
</tr>
<tr>
<td>5</td>
<td>Evolutionary Computing and Artificial Embryogeny</td>
<td>97</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Context: The PO-Plane</td>
<td>98</td>
</tr>
<tr>
<td>5.3</td>
<td>Evolutionary Computing</td>
<td>99</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Genetic Algorithms</td>
<td>100</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Genetic Programming</td>
<td>100</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Evolutionary Computing for Dynamic Optimization</td>
<td>101</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Self-Replicating and Self-Reproducing Code</td>
<td>102</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Self-Modifying Code</td>
<td>102</td>
</tr>
<tr>
<td>5.3.6</td>
<td>Indirect Encodings in Evolutionary Computing</td>
<td>103</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Approaches Based on Gene Expression</td>
<td>103</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Chemical Computing Models and Evolution</td>
<td>104</td>
</tr>
<tr>
<td>5.4</td>
<td>Embryology</td>
<td>105</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Embryonics</td>
<td>105</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Artificial Embryogeny</td>
<td>107</td>
</tr>
<tr>
<td>5.5</td>
<td>Discussion: EC and Embryology: Common Synergies</td>
<td>108</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusion</td>
<td>109</td>
</tr>
<tr>
<td>6</td>
<td>A Survey of Message Diffusion Protocols in Mobile Ad Hoc Networks</td>
<td>115</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>115</td>
</tr>
<tr>
<td>6.2</td>
<td>Broadcast and Dissemination Protocols</td>
<td>116</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Categorization by the used information</td>
<td>117</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Categorization by strategy</td>
<td>118</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Categorization by media access</td>
<td>119</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Survey of existing algorithms</td>
<td>120</td>
</tr>
<tr>
<td>6.2.5</td>
<td>General comparison and discussion</td>
<td>124</td>
</tr>
<tr>
<td>6.3</td>
<td>Routing Approaches for Intermittently Connected Networks</td>
<td>124</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>8.12.2 Capacity and SINR estimation</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>8.12.3 Power estimation</td>
<td>183</td>
<td></td>
</tr>
<tr>
<td>8.13 The Stieljes Transform approach</td>
<td>183</td>
<td></td>
</tr>
<tr>
<td>8.13.1 A theoretical application example</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>8.13.2 A wireless communication example</td>
<td>186</td>
<td></td>
</tr>
<tr>
<td>9 Tools from Physics and Road-traffic Engineering for Dense Ad-hoc Networks</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td>9.1 Introduction</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td>9.2 Determining routing costs in dense ad-hoc networks</td>
<td>193</td>
<td></td>
</tr>
<tr>
<td>9.2.1 Costs derived from capacity scaling</td>
<td>193</td>
<td></td>
</tr>
<tr>
<td>9.2.2 Congestion independent routing</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>9.2.3 Costs related to energy consumption</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>9.3 Preliminary</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>9.4 Directional Antennas and Global Optimization</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td>9.5 User optimization and congestion independent costs</td>
<td>199</td>
<td></td>
</tr>
<tr>
<td>9.6 Geometry of minimum cost paths</td>
<td>199</td>
<td></td>
</tr>
<tr>
<td>9.6.1 The function $U$ has the same sign over the whole region $R$</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>9.6.2 The function $U$ changes sign within the region $R$</td>
<td>202</td>
<td></td>
</tr>
<tr>
<td>9.7 User optimization and congestion dependent cost</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>9.8 Numerical Example</td>
<td>206</td>
<td></td>
</tr>
<tr>
<td>9.9 Conclusions</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td>9.10 Appendix: Mathematical Tools</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>10 Historical-Interpretive Considerations about Money as a Unit of Value and Scale-Dependent Phenomenon</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>10.1 Introduction</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>10.2 Historical Background</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>10.3 Community currencies working framework</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>10.4 Economics of sharing</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td>10.5 Technology as a mediator in the use of money and Empirical Business Models</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td>10.6 Further research applied to BIONETS</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>10.7 References</td>
<td>224</td>
<td></td>
</tr>
<tr>
<td>11 Eigenvector based reputation measures</td>
<td>225</td>
<td></td>
</tr>
<tr>
<td>11.1 Introduction</td>
<td>226</td>
<td></td>
</tr>
<tr>
<td>11.1.1 Estimation of trust</td>
<td>226</td>
<td></td>
</tr>
<tr>
<td>11.1.2 Estimation of reputation</td>
<td>226</td>
<td></td>
</tr>
<tr>
<td>11.1.3 Graph based reputation measure</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>11.2 Application of graph based reputation in BIONETS</td>
<td>229</td>
<td></td>
</tr>
<tr>
<td>11.3 Asynchronous approach</td>
<td>229</td>
<td></td>
</tr>
<tr>
<td>11.4 Aggregation/Decomposition Approach</td>
<td>230</td>
<td></td>
</tr>
<tr>
<td>11.4.1 Block-diagonal case</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>11.4.2 Full aggregation method (FAM)</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>11.4.3 Partial aggregation method (PAM)</td>
<td>233</td>
<td></td>
</tr>
<tr>
<td>11.4.4 BlockRank Algorithm (BA)</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td>11.4.5 Fast Two-Stage Algorithm (FTSA)</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td>11.4.6 Distributed PageRank Computation (DPC)</td>
<td>236</td>
<td></td>
</tr>
</tbody>
</table>
11.4.7 Discussion of the A/D methods .................................................. 237
11.5 Personalized approach ................................................................. 238
  11.5.1 Scaled Personalization ............................................................ 238
  11.5.2 Relation to the A/D approach ................................................. 240
11.6 Conclusions .............................................................................. 240

12 Network Formation Games
Giovanni Neglia .................................................................................. 245
12.1 Introduction ................................................................................ 245
12.2 Purpose and outline .................................................................... 246
12.3 Approaches ................................................................................ 246
12.4 Main Concepts in Network Formation Games ............................... 247
  12.4.1 Nash Equilibrium ................................................................... 247
  12.4.2 Other Equilibria for Coordination requirement ....................... 247
  12.4.3 Value Function and Allocation Rule ....................................... 248
  12.4.4 Price of anarchy/stability ........................................................ 248
12.5 Applications to Computer Networks ........................................... 249
  12.5.1 Local Connection Games ......................................................... 249
  12.5.2 Global Connection Games ....................................................... 250
  12.5.3 Overlay Specific Games .......................................................... 251
# DOCUMENT HISTORY

## Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Date</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Draft</td>
<td>29 May 2007</td>
<td>Daniele Miorandi, CN</td>
</tr>
<tr>
<td>0.5</td>
<td>Almost complete draft</td>
<td>15 July 2007</td>
<td>Eitan Altman, INRIA</td>
</tr>
<tr>
<td>0.9</td>
<td>Complete draft</td>
<td>29 July 2007</td>
<td>Paolo Dini, LSE</td>
</tr>
<tr>
<td>1</td>
<td>Revision of Complete draft</td>
<td>2 August 2007</td>
<td>Eitan Altman, INRIA</td>
</tr>
<tr>
<td>2</td>
<td>New version completed after a new round of reviews</td>
<td>15 September 2008</td>
<td>Eitan Altman, INRIA</td>
</tr>
</tbody>
</table>

## Summary of Changes

<table>
<thead>
<tr>
<th>Version</th>
<th>Section(s)</th>
<th>Synopsis of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>new cls file used</td>
<td>Inclusion</td>
</tr>
<tr>
<td>0.5</td>
<td>All Chapters of Part 1 and 2</td>
<td>Revised Versions Included</td>
</tr>
<tr>
<td>0.9</td>
<td>All Chapters of Part 3</td>
<td>Revised Versions Included</td>
</tr>
<tr>
<td>1</td>
<td>Including Index, Final corrections</td>
<td></td>
</tr>
</tbody>
</table>
Preface

At the hardware and lower stack level the BIONETS System is composed of a local network of mobile U-nodes that are connected among themselves but disconnected from any IP network or backbone, plus any number of sensors or T-nodes. 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 network and sensors.

The nature of the communication technology addressed in BIONETS research makes it necessary to examine both the lower technology-centric levels as well as the upper user-centric levels. Whereas the lower 2 levels in the table below (protocols and services) can benefit from biological, physical, and mathematical models, the upper 2 levels (services and users) cannot only benefit from social science paradigms, but actually require them to achieve a meaningful and sustainable integration of a BIONETS instance and its users.

<table>
<thead>
<tr>
<th>Level of BIONETS instance</th>
<th>Paradigms</th>
</tr>
</thead>
<tbody>
<tr>
<td>User level</td>
<td>Social science, business models</td>
</tr>
<tr>
<td>Service level</td>
<td>Biology, physics, mathematics, business models, social science</td>
</tr>
<tr>
<td>Protocol level</td>
<td>Biology, physics, mathematics</td>
</tr>
</tbody>
</table>

The above statement is based on several decades of research in the very challenging field of socio-economic analysis and business models for new media, carried out by diverse institutions all over the world. The recurring problem is how to generate a revenue model based on content sharing and distribution when the prevailing and seemingly unstoppable trend is towards free content. The overwhelming consensus is that the social dimension of technology is becoming increasingly important in generating a new kind of value. This can be seen, for example, in the Open Source movement, or in the growing Web 2.0 phenomenon where new channels and modes of expression that enable the formation of communities are continually being created (YouTube, Flickr, blogging, eBay, FaceBook, iGoogle, etc).

From the point of view of the 2020 Vision of Communications that motivated the FET call under which BIONETS is funded, 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)
• the long-term memory function that the sensors can support, which is similar to the pheronome trails of an ant colony and that affords the system with something equivalent to a distributed, sub-symbolic intelligence.

Clearly the architecture of the network 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 is investing 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. 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 a Bionet.

Performance optimisation and adaptability are therefore necessary to the realisation of the BIONETS vision. However, they are not sufficient. The realisation of the 2020 Vision of Communications necessitates an in-depth understanding of the needs and behaviour of the users if we want to develop an economically sustainable model of BIONETS-style communications. This is why Social Science is important in BIONETS. In other words, social science can help us understand what BIONETS applications should adapt to, thereby contributing to the definition of an appropriate fitness function.

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 articles 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). These paradigms will then be mapped to different parts of the project in the research that continues.

E Altman, P Dini, D Miorandi, D Schreckling
Chapter 1

Introduction

This document is the fruit of the collection of paradigms and foundations that had been identified as being relevant to the design, analysis and optimization of autonomous wireless networks, in general, and to the Bionets architecture and services, in particular. The document contains three parts: paradigms from biology, from physics and from social science. To summarize the motivation of studying these paradigms, we cite Kelly (1991) [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 the applicability to Bionets architecture and services.

1.1 Biologically inspired paradigms.

Machine Learning and control under uncertainty. Stochastic search algorithms have been widely researched and used in the past decades for solving complex optimization 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 games Evolutionary games provide a theoretical framework to understand and predict the evolution of services, of protocols and of the architecture of decentralized 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 as rules of evolution of protocols, architecture and services can be engineered so as to achieve desired stability and efficiency objectives. In our survey 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.

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. As a consequence 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. In addition to introductory material the article also provides recent references that support this perspective. The main areas of application within BIONETS for this work is in gene expression-based computing and security.

**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 among simpler units. First, the authors review Evolutionary Computing (EC) techniques, highlighting their potential application to the automated optimization 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.

**Message Diffusion Protocols in Mobile Ad Hoc Networks** The chapter surveys routing protocols in mobile sparse ad-hoc networks. Due to low connectivity, direct paths do not exist between a source and a destination, and end-to-end communication has to rely on mobility of terminals which relay packets. Tools from epidemiology have been used to analyze 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.

### 1.2 Paradigms Related to 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. Applications to dense networks are discussed.

**Scale-free networks.** Scale-free distributions are known to appear often in various phenomena arising 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 arising in organisational networks (usually expressed in terms of degree distribution of a graph representing the system’s relationships). This has been proven 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 nodes location in a network, which gives rise, in various cases, to fractal-like behaviour. The survey reviews methods and tools for characterizing and analyzing such cases, providing a useful model for (i) analyzing various issues related to BIONETS disappearing networks (ii) providing a set of conditions for the arising of such structures, which are known to show peculiar properties (robust yet fragile).

**Spatial models from Optics, Electrostatics and Road traffic.** 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 Chapter 6 that 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.

**Random Matrix Theory and free probability** In many applications of wireless communications, we encounter matrices whose entries are random. Examples are 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 grow. The Theory of random matrices allows us to derive explicit expressions for those limits and thus provide a powerful tool for computing performance measures of systems with a large number of mobiles. This approach is based in part on a recent theory called Free Probability which is surveyed as well.

### 1.3 Paradigms from Social Sciences.

**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.

**Distributed approaches to graph 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 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 personalization methods which use local information.

**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 current Internet the network structure arises from interactions of agents at different levels. Internet Service Providers (ISPs) and different organizations 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.
Bibliography


Part I

Biological Inspired Paradigms
Chapter 2

Machine Learning for Intelligent Optimization

M. Brunato

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.

2.1 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 Traveling 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. 2.2 defines the application domain of optimization meta-heuristics. Sec. 2.3 provides a review of the basic meta-heuristic techniques. Sec. 2.4 introduces the framework of reactive search and presents a survey of recent developments. Applications of such techniques are explored in Sec. 2.5. Finally, Sec. 2.6 discusses the relevance of the proposed techniques within the BIONETS project.

2.2 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 $X$; an element $X \in X$, 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 $f: X \rightarrow \mathbb{R}$. 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 $X \in X$ 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 $(X, f)$ is called an optimization problem. If the configuration space $X$ 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 $X$ 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 ($\text{MAXSAT}$) problem (see [25] for a classical survey), where $n$ Boolean variables $x_1, \ldots, x_n$ are given. An atom is defined as either a variable $x_i$ or the negation of a variable $\overline{x}_i$, 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 $X = \{0, 1\}^n$ (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 \((X, f)\) describing the problem, and outputs an element \(X \in X\) 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 \(X\) 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. 2.2.1–2.2.2.

2.2.1 Neighborhood 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 neighborhood for every configuration. Consider the MAXSAT case: given a configuration \(X \in X\), let its neighborhood be defined as all configurations that differ from it by at most one bit:

\[
N(X) = \{X' \in X | H(X, X') \leq 1\},
\]

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

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

An immediate advantage of considering this neighborhood 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 neighborhood 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 \(X\), the objective function can be expected to manifest some form of “regularity” with respect to it. In particular, the triplet \((X, \mathcal{N}, f)\) can be analyzed in terms of a landscape [35, 60] in order to study the dynamics of solving algorithms on particular problems.

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
November 5, 2008  D2.1.1 Survey on paradigms  BIONETS/wp2.1/0.1

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>(input) Search space description</td>
</tr>
<tr>
<td>$\mathcal{N}$</td>
<td>(input) Neighborhood structure</td>
</tr>
<tr>
<td>$f$</td>
<td>(input) Objective function</td>
</tr>
<tr>
<td>$t$</td>
<td>(local) Iteration counter</td>
</tr>
<tr>
<td>$X^{(t)}$</td>
<td>(local) Current configuration at iteration $t$</td>
</tr>
<tr>
<td>$\hat{X}$</td>
<td>(local) Best configuration found so far</td>
</tr>
</tbody>
</table>

1. function LocalSearch ($X$, $\mathcal{N}$, $f$)
2. $t \leftarrow 0$
3. $X^{(0)} \leftarrow$ choose an element in $X$
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 $\mathcal{N}(X^{(t-1)})$
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

Figure 2.1: The basic Local Search framework: the algorithm moves between neighboring configurations

this principle. The basic structure of a local search algorithm is shown in Fig. 2.1: the algorithm repeatedly moves from a configuration to a neighboring one (the loop in lines 2.2.1–2.2.1), storing the best configuration found so far (lines 2.2.1–2.2.1). Heuristics differ in the choice of the initial configuration (line 2.2.1 of Fig. 2.1), of the subsequent neighbors (line 2.2.1), and the continuation condition (line 2.2.1).

2.2.2 Large-scale structure

In the previous section we have seen how a local search optimization algorithm can move between neighboring 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 modeled 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 neighborhood becomes clearly counterproductive. Therefore, a diversification, or exploration strategy is needed. For example, 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: rather than restarting elsewhere, 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 behavior 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. 2.3.
2.3 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.

2.3.1 Variable Neighborhood Search

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

The seminal idea of the Variable Neighborhood Search (VNS) technique [24] is to consider a set of predefined neighborhoods, and aim at using the most appropriate one during the search, as illustrated in Fig. 2.2, where the possible neighborhoods are represented as concentric circles around configurations, and the selected one is bold.

Many schemes for using the set of different neighborhoods in an organized way are possible [27]. Variable Neighborhood Descent (VND) uses a default neighborhood first; other neighborhoods are ranked in some order of importance and are used only if the default neighborhood 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 neighborhoods 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 neighbor 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. 2.3.2) 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.

Figure 2.2: Variable neighborhood search: the used neighborhood (bold circle around the current configuration) varies along the search trajectory.
2.3.2 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 behavior 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 neighbor:

$$
X' \leftarrow \text{random element in } N(X(t))
$$

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

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 analyzed for different problems [2, 16, 17].

2.3.3 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 $n$: $X = \{0, 1\}^n$. Given the current configuration $X(t)$, in Tabu Search only a subset $N_A(X(t)) \subset N(X(t))$ of neighbors is allowed, while the other neighbors are prohibited. The general way of generating the search trajectory that we consider is given by:

$$
X(t+1) = \text{BESTNEIGHBOR}(N_A(X(t)))
$$

$$
N_A(X(t+1)) = N(X(t+1)) \cap \text{ALLOW}(X(0), \ldots, X(t+1))
$$

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 $X(0), \ldots, X(t+1)$. 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 $X \in X$, let its neighbors 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. 2.3: 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 neighbors that are 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. 2.3 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 2.4.3 we shall illustrate some possible solutions to these problems.

### 2.3.4 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...
Figure 2.4: 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.
sequence of basic operations: the fitness of each individual is computed, then some individuals are chosen by a random selection process that favors elements with a high fitness function (line 2.3.4); a cross-over combination is applied to randomly selected survivors in order to combine their features (lines 2.3.4–2.3.4), then some individuals undergo a random mutation (lines 2.3.4–2.3.4). 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 $\text{MAXSAT}$, 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

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

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.

### 2.3.5 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.

Although model-based techniques can be used in both discrete and continuous domains, the latter case better supports intuition. In Fig. 2.5 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 neighborhood 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 modeling 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 modeling promising solutions by estimating their probability distribution and using the model to guide the exploration of the search space.

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
Figure 2.6: 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.

in Fig. 2.6, maintains a probabilistic generative model with parameters $p = (p_1, \ldots, p_n) \in [0, 1]^n$. This model is used to generate a population of candidate solutions $P$ (line 2.3.5). A selection of the fittest solutions (line 2.3.5) is used to modify the probability estimates of the solution’s components, which are incrementally updated by a moving average (lines 2.3.5–2.3.5).

Another algorithm in the EDA framework is Mutual-Information-Maximizing Input Clustering (MIMIC) [19]. Given a function $f$ to minimize within configuration space $X$, the technique tries to set a convenient threshold $\theta$ and to estimate the distribution $p^\theta$ of solutions whose objective value is lower than $\theta$. For selecting the proper threshold (notice that if $\theta$ 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.

2.4 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. 2.2, and applied to the heuristics described in Sec. 2.3. 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
Figure 2.7: Parameter tuning in optimization heuristics can depend on a human researcher; Reactive Search aims at machine learning-based automatic tuning.

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.

The context is represented in Fig. 2.7, where the researcher acts as an intelligent feedback channel that is able to adjust parameters according to his observations about the algorithm’s behavior, 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 offline manner as in the human researcher’s case, so that the learning component shown in Fig. 2.7 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 metaheuristics are surveyed.

2.4.1 Variable Neighborhood 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 neighborhood ordering for VND is presented in [32]. Ranking of the different neighborhoods depends on their observed benefits in the past and is dynamically changed during the search.

2.4.2 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 $T_{\text{found}}$ 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 $\gamma$ 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, $\gamma$ 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 (2.1) 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.

2.4.3 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 signaled 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].

2.4.4 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.

2.5 Biology-inspired applications

In this section we present two examples of reactive search applied to biology-inspired scenarios. In the first case (Section 2.5.1) 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 2.5.2) the distribution of local search algorithms on several machines is studied in terms of a tradeoff between the rate of knowledge exchange (performed in an epidemic, peer-to-peer fashion) and the quality of the solution found.

2.5.1 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 meta-heuristics, 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.

2.5.2 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 behaviors) 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
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 tradeoff 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. 2.8 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 $10^{-2}$).
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 tradeoff between the amount of information which can be effectively exchanged and the quality of the solution is being studied.

2.6 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. 2.7) to let the algorithm observe its own evolution in time and “react” appropriately by modifying its own behavior 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 2.5) can be particularly suited for application in such context.

2.7 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 tradeoff between the quality of the results and the computational burden of maintaining and searching the history-dependent data structures that lie at their foundations.
Bibliography


Chapter 3

Evolutionary Games

H. Tembine, E. Altman, Y. Hayel, R. El-Azouzi

Abstract. Evolutionary games have developed in biological sciences to study equilibrium behavior (called Evolutionary Stable Strategies – ESS) among large populations. ESS are more adapted than the standard Nash equilibrium to predict the behavior 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.

3.1 Introduction

The evolutionary games formalism is a central mathematical tools developed by biologists for predicting population dynamics in the context of interactions between populations. This formalism identifies and studies two concepts: The ESS (for Evolutionary Stable Strategy), and the Replicator Dynamics [13].

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

1. if an ESS is reached, then the proportions of each population do not change in time.

2. at ESS, the populations are immuned from being invaded by other small populations. This notion is stronger than Nash equilibrium. Indeed, in the Nash equilibrium concept, it is only requested that a single user would not benefit by a change (mutation) of its behavior.

ESS has first been defined in 1972 by M. Smith [21], who further developed it in his seminal text Evolution and the Theory of Games [22], followed shortly by Axelrod’s famous work [4].

\footnote{The Nash equilibrium is the main solution concept in non-cooperative games where each one of several players has its own utility which it wishes to optimize. The Nash equilibrium is a multipolicy (i.e. a policy set that consists of a given policy assignment to each user), with the property that no user can benefit by a deviating unilaterally from its assignment. This is in contrast with the notion of a Pareto optimal solution defined as a multipolicy for which it is impossible to improve the utility of one player (by a deviation of one or more players) without decreasing the utility of at least one other player.}
Although ESS has been defined in the context of biological systems, it is highly relevant to engineering as well (see [29]).

Recently, evolutionary game theory has gained interest among social scientists [10].

In the biological context, the replicator dynamics is a model for the change of the size of the population(s) that biologists observe, where as 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 behaviors: 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 modeling 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.

3.2 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 neighboring 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 who adopts 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. 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 $q$.

### 3.3 Evolutionary Stable Strategies

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

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

(3.1)

A strategy $q$ is said to be ESS if for every $p \neq q$ there exists some $\varepsilon_y > 0$ such that (3.1) holds for all $\varepsilon \in (0, \varepsilon_y)$.

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

$$J(q, q) > J(p, q)$$

(3.2)

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,

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

(3.3)

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 (3.2) or (3.3), see [30, Proposition 2.1] or [12, 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.

### 3.4 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. 3.1 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 3.1. The utilities represent the following:

1. An encounter D–D results in a peaceful, equal-sharing of the food which translates to a fitness of 0.5 to each player.
2. 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.

3. 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.

![Figure 3.1: A H–D game in matrix form](image1)

![Figure 3.2: Generalized H-D game](image2)

**Classification of equilibria in the HD game**

A more general description of H–D games is available in [18, 14]. 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 [7] of the HD game given in Figure 3.2, 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. $a = u + v$ is the unique interior Nash equilibrium where $A_{ij} = J(i,j)$, $i,j \in \{H,D\}$, $u = A_{12} - A_{22}$, $v = A_{21} - A_{11}$.

Remark 3.4.1 (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.

3.5 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 [24]).

To be more precise, consider the case of a finite number $N$ of strategies. Let $x(t)$ be the $N$ dimensional vector whose $i$th element $x_i(t)$ is the population share of strategy $i$ at time $t$. Thus $\sum x_i(t) = 1$ and $x_i(t) \geq 0$. 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 $J(i,x(t)) = \sum J(i,j)x_j(t)$.

Then the replicator dynamics is defined as

$$
\dot{x}_i(t) := \frac{dx_i(t)}{dt} = x_i(t)K \left( J(i,x(t)) - \sum_j x_j(t)J(j,x(t)) \right)
$$

(3.4)

$$
= x_i(t)K \left( \sum_j x_j(t)J(i,j) - \sum_j \sum_k x_j(t)J(j,k)x_k(t) \right)
$$

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 [25]. It has also been studied in the context of the association problem in wireless networks in [26].

3.6 Replicator dynamics with delay

In (3.4), 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 $\tau$ time later. This gives the following dynamics:

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

(3.5)
\[ x_i(t)K \left( \sum_j x_j(t-\tau)J(i,j) - \sum_{j,k} x_j(t)J(j,k)x_k(t-\tau) \right) \]

where \( K \) is some positive constant. We should mention that the delay \( \tau \) 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.

### 3.7 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 \([6]\) dynamics, the fictitious play dynamics and gradient methods \([20]\). See also \([24, 28]\).

We finally mention the logistic differential equation \([12]\):

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

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.

### 3.8 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 necessarily symmetric.

#### 3.8.1 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 "neighbor").

We assume the mobiles move frequently and they have a packet to send in each time slot. A mobile decides 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 two neighbors 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. Consider two anonymous 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 $\tau_T$. The packet transmission is successful if Player II does not transmit (stays quiet) in that given time slot and its delay is $\tau_S$, 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 $\tau_T$. The interaction is represented in figure 3.3.

### 3.8.2 Delay impact on the stability

The one-shot game 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

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

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 (3.6) if $2K\Delta(1 - \Delta)\tau_T < \pi$. The proof for the case of $K = 1$ can be found in [28]. The adaptation to any $K$ can be found in [3].

The trajectories of the population using the strategy $T$, as a function of time is represented in Fig.3.4. We evaluate the stability varying the delay $\tau_T$ in the replicator dynamic. When the delay $\tau_T$ is large, the trajectory oscillates rapidly and the amplitude is seen to be greater than the equilibrium point and the system is unstable. Note that the stability condition is independent of $\tau_S$.

![Figure 3.4: Effect of $\tau_T$ on velocity and stability of replicator dynamic. $\Delta = 1/3$](image)
3.9 Application to transport protocols

We summarize below a study in [3] 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 looses data units has to retransmit them later. In the absence of adaptation to the congestion, the on going 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 $\alpha$ packets every round trip time and when it detects congestion it decreases the window size to $\beta$ times its value. The constants $\alpha$ and $\beta$ 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 modeled by an AIMD behavior where $\alpha$ and $\beta$ are not constant anymore : $\alpha$ and $\beta$ 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 analyzed 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 depend 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 performances 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 $\alpha_i$ and a multiplicative decrease factor $\beta_i$. Let $x_i(t)$ be the transmission rate of connection $i$ at time $t$. We consider the following simple model for competition.

1. The RTT (round trip times) are the same for all connections.

2. 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).

3. Losses occur whenever the sum of rates reaches the capacity $C$: $x_1(t) + x_2(t) = C$.

4. 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 $\beta_i < 1$.

5. As long as there are no losses, the rate of connection $i$ increases linearly by a factor $\alpha_i$.

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 $J_i = T \text{hp}_i - \lambda R$ for connection $i$; it is the difference between the throughput $T \text{hp}_i$ and the loss rate $R$ weighted by the so called tradeoff parameter, $\lambda$, 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 $\lambda$.

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

$$T \text{hp}_1 = \frac{1 + \bar{\beta}_1}{2} \times \frac{\alpha_1 \bar{\beta}_2}{\alpha_1 \bar{\beta}_2 + \alpha_2 \bar{\beta}_1} \times C.$$
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 \((\alpha_H, \beta_H)\) and \( D \) to \((\alpha_D, \beta_D)\) 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 \( \lambda \) and the ESS is polymorphic i.e the two versions of TCP would coexist.

A stability analysis is carried out in [3] and an oscillatory behavior is identified in the non-stable case.

### 3.10 Conclusions

We presented in this paper some basic notions of the theory of evolutionary games. We highlighted the advantages of this paradigm with respect to traditional non-cooperative 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.
Bibliography


Chapter 4

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

P. Dini, D. Schreckling

4.1 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 de-
The work in Task 2.1.1.4 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. Table 4.1 summarises these facts for a species whose individuals live for a few decades.

<table>
<thead>
<tr>
<th>Process</th>
<th>Mechanism</th>
<th>Result</th>
<th>Time Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolution</td>
<td>genetic operators</td>
<td>phylogenetic trees</td>
<td>millions of years</td>
</tr>
<tr>
<td></td>
<td>natural selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Morphogenesis</td>
<td>gene expression</td>
<td>adult organism</td>
<td>years</td>
</tr>
<tr>
<td></td>
<td>cell differentiation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metabolism</td>
<td>gene expression</td>
<td>structure and behaviour</td>
<td>days-hours-minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cognition</td>
<td>neuronal network</td>
<td>thought</td>
<td>milliseconds</td>
</tr>
</tbody>
</table>

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]. 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 [3], 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 [35] supports this perspective and is in fact ahead of our results. 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 [3] [2] [4] [5], 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.

![Diagram](image.png)

**Figure 4.1:** **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 [4] [5]. 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 [6] 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 [7] [8] [9]. 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.

4.2 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.

4.2.1 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 [10] [11] [12].

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 $A$.

### 4.2.2 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) $a + (b + c) = (a + b) + c, \forall (a, b, c) \in R$
Table 4.1: Truth table for the $\mathbb{Z}_2$ field

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.1: Truth table for the $\mathbb{Z}_2$ field

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

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 4.1.

### 4.2.3 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 \(^\text{'co'}\) in \(^\text{'coset'}\) stands for \(^\text{`complementary'}\); 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

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

(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 \( a \mod 5 = b \mod 5 = 0 \)? If yes, \( a \) and \( b \) belong to \( S \), which is clearly the infinite set

\[
S = \{..., -15, -10, -5, 0, 5, 10, 15, ...\}.
\]

(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 [11] defines it in general as follows:

\( \mathbb{Z}_n \) is the set of equivalence classes determined by the equivalence relation \( = \text{ (mod n)} \).

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

\[
\mathbb{Z}_5 = \{0, 1, 2, 3, 4\},
\]

(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:

\[
S = \{..., -15, -10, -5, 0, 5, 10, 15, ...\} \rightarrow 0
\]

\[
S + 1 = \{..., -14, -9, -4, 1, 6, 11, 16, ...\} \rightarrow 1
\]

\[
S + 2 = \{..., -13, -8, -3, 2, 7, 12, 17, ...\} \rightarrow 2
\]

\[
S + 3 = \{..., -12, -7, -2, 3, 8, 13, 18, ...\} \rightarrow 3
\]

\[
S + 4 = \{..., -11, -6, -1, 4, 9, 14, 19, ...\} \rightarrow 4
\]

(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

$$\theta: < R, +, \cdot > \rightarrow < S, \oplus, \odot >$$

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

$$\theta(a + b) = \theta(a) \oplus \theta(b) \quad (2.6a)$$

$$\theta(a \cdot b) = \theta(a) \odot \theta(b) \quad (2.6b)$$

![Figure 4.2: The four basic kinds of mappings](image)

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

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

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**.

### 4.2.4 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 4.3: **Showing how homorphisms are not 1-1 and Onto**
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

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

The **image** is a subring of \( S \) defined by

\[
\text{Im}(\theta) = \{ s \in S : s = \theta(r) \text{ for some } r \in R \}.
\] (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 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 \( R \) iff

\[
\begin{align*}
(a) & \forall s_1, s_2 \in S, s_1 - s_2 \in S \\
(b) & \forall s \in S \text{ and } r \in R, rs, sr \in S.
\end{align*}
\] (2.9a)

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

\[
\begin{align*}
(I + x) + (I + y) &= I + (x + y) \\
(I + x)(I + y) &= I + xy
\end{align*}
\] (2.10a,b)

### 4.2.5 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 [10].

A **zero divisor** in a ring \( R \) is a non-zero element \( a \in R : \exists (b \neq 0) \in R \text{ with } ab = 0 \). 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

1. $R$ is a subring of $F$
2. 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.

### 4.2.6 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 $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m \in F[x]$ 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 [13]. 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 future work) 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
\[
S = \frac{R}{I} = \frac{F[x]}{[f(x)]},
\]
(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:
\[
a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5.
\]
(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
\[
I + r(x),
\]
(2.13)
where \( r(x) \) is the remainder we have been talking about. Now suppose that

\[
f(x) = b_0 + b_1x + b_2x^2 + ... + b_nx^n
\]  

(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 \( F[x] \) \( \text{GCD}(r(x), f(x)) = 1 \). One of the algebra basics we glossed over is the result that, in \( F[x] \), \( \exists s(x), t(x) : \)

\[
\begin{align*}
s(x)r(x) + t(x)f(x) &= \text{GCD} \\
s(x)r(x) + t(x)f(x) &= 1 & \text{in } F[x] 
\end{align*}
\]  

(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)),

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

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 \( \theta \) 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.

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 \( \alpha \). 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.
Figure 4.4: Finding the root of \( f(x) \)

in \( F \). Because \( x \) is of degree smaller than \( f(x) \) it is one of the remainders, and therefore it is an element of \( S \):

\[
\theta(x) = x + I = \alpha.
\]  

(2.18)

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

\[
\begin{align*}
  f(\alpha) &= b_0 + b_1\alpha + b_2\alpha^2 + \ldots + b_m\alpha^m \\
  &= b_0 + b_1(x + I) + b_2(x + I)^2 + \ldots + b_m(x + I)^m \\
  &= b_0 + (b_1x + I) + (b_2x^2 + I) + \ldots + (b_mx^m + I) \\
  &= I + (b_0 + b_1x + b_2x^2 + \ldots + b_mx^m) \\
  &= I + f(x) \\
  &= I \\
  &= 0
\end{align*}
\]  

(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 \( \theta \) into the image of \( r(x) \):

\[
\begin{align*}
  I + r(x) &= I + (a_0 + a_1x + a_2x^2 + \ldots + a_{m-1}x^{m-1}) \\
  &= a_0 + (a_1x + I) + (a_2x^2 + I) + \ldots + (a_{m-1}x^{m-1} + I) \\
  &= a_0 + a_1(x + I) + a_2(x + I)^2 + \ldots + a_{m-1}(x + I)^{m-1} \\
  &= a_0 + a_1\alpha + a_2\alpha^2 + \ldots + a_{m-1} - \alpha^{m-1} \\
  &= r(\alpha)
\end{align*}
\]  

(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 $\alpha$ play the role of basis vectors. We recognise in this the same structure of the complex field $C$, 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 $\mathbb{R}$.

4.3 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 emphasized 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.

4.3.1 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 4.3.2 and 4.3.3. However, for the experienced reader who is familiar with the basic concepts of these logics we recommend to proceed directly to sub-section 4.3.2.

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 $\text{and}$ and $\text{or}$, 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 $\bot$ for the values $\text{true}$ and $\text{false}$ as it is usually done in classical two-valued logic. Based on these assumptions we define propositional atomic formulas and propositional formulas.

**Definition 4.3.1** An atomic formula is a propositional letter, $\top$ or $\bot$. 
Table 4.2: Definition of propositional logic connectives

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

**Definition 4.3.2** The set of propositional formulas is the smallest set $P$ such that

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

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 $T = \{0, 1\}$. Looking at definition 4.3.2 we additionally have to define the operation on set $T$ represented by the operation symbols $\neg$ and $\circ$.

There are some similarities and differences between these definitions and the starting point of abstract algebra. $P$ is analogous to an infinite set of variables whose values are taken from the field $Z_2 = T$. In abstract algebra we define more complicated 'objects' over a particular field. These objects are called polynomials. So we can have polynomials over $Z_2$ or over $Z, N, Q, R, C$. 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 $\neg : T \rightarrow T$ by $\neg(0) = f$ and $\neg(1) = t$. Defining the binary connectives is more complicated as there are 16 possible mappings. We can enumerate the definitions of all connectives in Table 4.2.

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

**Definition 4.3.3** A Boolean valuation is a mapping $v$ from the set of propositional formulas to the set $T$ meeting the conditions

1. $v(\top) = 1; v(\bot) = 0$
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 4.2 can be compared to Table 4.1. Although at first these seem very different, it turns out that all the binary connectives shown in Table 4.2 can be expressed in terms of only two connectives, \( \land \) (multiplication) and \( \not\equiv \) (addition mod 2). The additional structure shown by Table 4.2 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 4.3.4**

- 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 formulas 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 4.2.

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 \( \neg(\neg(P \subset Q) \land \neg R) \). 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.

\[
v(\neg(\neg(P \subset Q) \land \neg R)) = \neg v(\neg(P \subset Q) \land \neg R) \\
= \neg(\neg v(P \subset Q) \land \neg v(\neg R)) \\
= \neg(\neg (v(P \subset Q) \land \neg v(\neg R))) \\
= \neg(\neg (1 \land \neg 0)) \\
= \neg(\neg 1) \\
= \neg 0 \\
= 1.
\]

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 formulas 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 4.3.5** A first-order language \( L(\mathcal{R}, \mathcal{F}, \mathcal{C}) \) 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 arity\(^1\) 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 \( \mathcal{C} \) 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 formulas, which are analogous to propositional logic formulas. For this purpose we have to introduce variables. They represent elements of a set which we will later call domain.

**Definition 4.3.6** The family of terms of \( L(\mathcal{R}, \mathcal{F}, \mathcal{C}) \) is the smallest set meeting the following conditions:

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

Terms are closed if they contain no variables.

Based on the terms defined above we can now define formulas. You will recognise that the definition is very similar to the definition of formulas 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 formulas first.

**Definition 4.3.7** An atomic formula of \( L(\mathcal{R}, \mathcal{F}, \mathcal{C}) \) is any string of the form \( R(t_1, t_2, \ldots, t_n) \) with \( R \in \mathcal{R} \) and \( t_1, t_2, \ldots, t_n \in \mathcal{T}(\mathcal{R}, \mathcal{F}, \mathcal{C}) \).

---

\(^1\)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)
Based on atomic formulas we define the family of formulas of $L(R,F,C)$.

**Definition 4.3.8** The family of formulas of $L(R,F,C)$ is the smallest set meeting the following conditions

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

By using the syntax defined above and defining $R$, $F$, and $C$ appropriately we can specify the same formulas 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 4.3.9** A model for the first-order language $L(R,F,C)$ is a pair $\mathcal{M} = < D, I >$ with $D$ is non-empty set, called a domain of $\mathcal{M}$

1. is a mapping, called an interpretation that associates
   1. To every $c \in C$, some member $c^I \in D$
   2. To every $f \in F$, some function $f^I : D^n \rightarrow D$
   3. To every $P \in R$, some relation $P^I \subseteq D^n$

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 4.3.10** Let $\mathcal{M} = < D, I >$ be a model for the language $L(R,F,C)$, and let $\mathcal{A}$ be an assignment in $\mathcal{M}$. To each $t \in T_{L(R,F,C)}$ we associate a value $t^I^\mathcal{A}$ in $D$ as follows:

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

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 $L(R,F,C)$. This can be done in analogy to propositional logic. We are also going to use the same propositional constants $\top$ and $\bot$ as well as the same connectives (denoted as $\circ$) as in propositional logic.
Definition 4.3.11 Let $\mathcal{M} = <D, I>$ be a model for $L(R, F, C)$. Let $A$ be an assignment in this model. To each formula $\Phi$ of $L(R, F, C)$ we associate a value $\Phi^I,A \in T$ as follows:

1. For atomic cases
   
   $[P(t_1, t_2, \ldots, t_n)]^I,A = t \iff t_1^I,A, t_2^I,A, \ldots, t_n^I,A \in P^I,$
   
   $\top^I,A = t,$
   
   $\bot^I,A = f.$

2. $[\neg X]^I,A = \neg[X^I,A].$

3. $[X \circ Y]^I,A = X^I,A \circ Y^I,A$

4. $[(\forall x)\Phi]^I,A = t \iff \Phi^I,B = t$ for every assignment $B$ in $\mathcal{M}$ that is an $x$-variant of $A.$

5. $[(\exists x)\Phi]^I,A = t \iff \Phi^I,B = t$ for some assignment $B$ in $\mathcal{M}$ that is an $x$-variant of $A.$

Here the assignment $B$ in the model $\mathcal{M}$ is an $x$-variant of the assignment $A$, if $A$ and $B$ 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 $L(\{R\}, \{\oplus\}, 0)$ with variables $x$ and $y$. We choose $\mathcal{M} = <N, I>$ as the model with $I$ defined as follows:

1. $\oplus^I(a, b) = a + b$, with $a, b \in N$, and

2. $R^I = \{(x, y) : x, y \in N, x > y\}$

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 [14].

4.3.2 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 4.3.1 we introduced the basic syntax and semantic of propositional logic. We defined Boolean evaluation and showed how a simple formula in propositional logic is valuated. If we want to generalize the Boolean valuation for arbitrary formulas with $n$ propositions we will have to investigate $2^n$ different interpretations.
This characteristic becomes a problem if we do not simply want to valuate propositional formulas but if we want to draw logical consequences from them.

**Definition 4.3.12** A propositional formula $X$ is a logical (propositional) consequence of a set $S$ of propositional formulas, 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 formulas one may use established proof procedures such as Hilbert systems or resolution. Logical consequences can also be interpreted as a means to find propositional formulas 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 formulas with the least number of connectives in it. This may be achieved by simply guessing formulas and proving their logical equivalence by valuation.

In section 4.2 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 formulas. As an example consider the atomic formula which consists of the proposition $P$. This formula can easily be extended with the propositional letter $\bot$ to $P \lor \bot$. Both formulas are equivalent as they have the same model. If we now merge all propositional formulas 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 formulas with model $M$. Obviously, we can group all propositional formulas based on the propositions in $P = \{P, R\}$ and obtain 16 equivalence classes. We call the set of these classes, set $E$.

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 4.3.13** A propositional formula $X$ is a tautology if $v(X) = 1$ for every interpretation $v$.

**Definition 4.3.14** A set $S$ of propositional formulas is satisfiable if there is some interpretation $v$: $v(s) = 1, \forall s \in S$.

Thus, the propositional formula $P \lor \neg P$ 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 $[\bot]$.

Intuitively, we now want to define a partial order on $E$. We define the order $\leq$ as follows:

$x \leq y \iff v(p) = v(P \land R), \forall P \in X$ and $\forall R \in Y$, with $X, Y \in E$ and for all interpretations $v$.

Why is this an intuitive definition? Consider the valuation of formula $P \land R$ with the interpretation $v$ with $v(P) = \top$ and $v(R) = \bot$. This yields $v(\top \land \bot) = v(\bot)$ which also means that $\bot \leq \top$ which would be an intuitive definition of a relation on the set $\{\bot, \top\}$. 


Figure 4.5: Hasse diagram of a Boolean lattice of propositional logic generated by \{P,R\}

We draw this partial order in figure 4.5\(^2\). Note that this figure is a Hasse diagram which is a graph representation for partially ordered sets (posets) \((H, \leq)\). Vertices \(a, b \in H\), \(a \neq b\) are only connected by an edge if \(a \leq b\) and there is no \(c \in H\) 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 \(H\).

**Definition 4.3.15** The infimum of a subset \(S\) of a partially ordered set \((H, \leq)\) is an element \(h \in H\) such that \(h = \inf(H) = \max(\{x \in H | \forall y \in H, x \leq y\})\).

**Definition 4.3.16** The supremum of a subset \(S\) of a partially ordered set \((H, \leq)\) is an element \(h \in H\) such that \(h = \sup(H) = \min(\{x \in H | \forall y \in H, y \leq x\})\).

If we now take a closer look at this poset \((\mathcal{E}, \leq)\) by drawing its Hasse diagram representation we will recognize two things.

First of all the poset is bounded by a least and a greatest element, \(\bot\) and \(\top\) respectively.

The second observation is not that obvious and requires a trained eye for propositional logic formulas. Take representatives \(X\) and \(Y\) from two arbitrary equivalence classes \([X]\) and \([Y]\). The application of the propositional connectives \(\lor\) and \(\land\) 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 \([Z] \in \mathcal{E}\).

---

\(^2\)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
To illustrate this we consider the arbitrarily picked equivalence classes \([P \lor R] \) and \([\neg P] \) which form the new set \([ [P \lor R], [\neg P]]\). We form the propositional formulas \( V = (P \lor R) \lor \neg P \) and \( W = (P \lor R) \land \neg P \).

Table 4.3 shows the possible interpretations of formulas \( V \) and \( W \).

<table>
<thead>
<tr>
<th>( v(P) )</th>
<th>( v(R) )</th>
<th>( v(P \lor R) )</th>
<th>( v(\neg P) )</th>
<th>( v(V) )</th>
<th>( v(W) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.3: All interpretations of supremum and infimum of \( v(P \lor R) \) and \( v(\neg P) \)

If we compare the truth values with table 4.2 we can verify that \( V \) is equivalent to \( \top \) and \( W \) is equivalent to \( P \not\subset R \). Their equivalence classes are again part of \( E \). Due to the fact that we used a Hasse diagram to display our partial order \((E, \leq)\) we can instantly see, that \( \top \) is supremum and \( P \not\subset R \) is infimum of set \([ [P \lor R], [\neg P]]\).

Here, we already see, that the order theoretic interpretation based on the partial order \( \leq \) finds its equivalance in an algebraic structure based on the two operators \( \lor \) and \( \land \) and vice versa.

Based on this observation and using the above Hasse diagram we may additionally verify that any three elements of \( E \) of the presented structure \((E, \leq)\) also have the characteristics of

1. associativity,
2. commutativity, and
3. absorption.

Thus the partial order \((E, \leq)\) 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 4.5.

Take three arbitrary elements from \( E \), e.g. the propositional formulas \( A = \neg R, B = \neg P, \) and \( C = P \not\subset R \). The distributivity of lattice \((E, \leq)\) or its correspondent \((E, \lor, \land)\) requires the following equivalence

\[
A \land (B \lor C) = (A \land B) \lor (A \land C), \text{ or}
\]

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 \( inf(\{A, C\}) = \bot \). Thus, it is \( inf(\{A, P \uparrow R\}) = P \downarrow R \) and \( sup(\{P \downarrow R, \bot\}) = P \downarrow R \) which proves distributivity for the chosen formulas.
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 formulas 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 [8, 15].

We generalize this result and give the definition for a Boolean algebra.

**Definition 4.3.17**

A Boolean algebra is a structure $B = < B, +, \cdot, 0, 1 >$ where the following system of equations is valid, and where $x, y, z \in B$:

\[
\begin{align*}
\begin{array}{ll}
x + (y + z) &= (x + y) + z \\
x + y &= y + x \\
x + (x \cdot y) &= x \\
x + (y \cdot z) &= (x + y) \cdot (x + z) \\
x + \overline{x} &= 1 \\
x \cdot \overline{x} &= 0
\end{array}
\end{align*}
\]

(associativity)

(commutativity)

(absorption)

(distributivity)

(existence of complement)

With this mathematical structure the Boolean algebra for propositional logic (PL) can be defined as the Boolean algebra model $B_{PL} = < E, \lor, \land, \neg, [\perp], [\top] >$.

You will realize that this algebra is only based on the operators $\lor$, $\land$, 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 $\lor$ and $\land$.

Summarizing 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 formulas into other propositional and equivalent formulas. These equivalence transformations have numerous areas of applications, such as integrated circuit optimization 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 analyze a structure. Consequentially, 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 algebraization. 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 algebraizations 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” [16].
From section 4.3.1 we already know that first-order logic is able to express formulas 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 algebraization 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 formulas. 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 $\Gamma$ be the first-order language $L(\mathcal{R}, \mathcal{F}, C)$ and let $< v_\kappa >_{\kappa<\alpha}$ a sequence of variables. Let $\Theta$ be a theory of $\Gamma$. We define $\mathcal{F}^\Gamma$ as the set of all formulas of $\Gamma$. We currently have to restrict our considerations to the set of formulas which does not contain the formulas $F \equiv_\Theta G$. Here relation $\equiv_\Theta$ denotes the equality relation which can be deduced from $\Theta$. 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 $\mathcal{B} = < B, +, \cdot, 0, 1 >$ we can define the following Boolean operations:

$$
(F/\equiv) + (G/\equiv) = F \lor G/\equiv \\
(F/\equiv) \cdot (G/\equiv) = F \land G/\equiv \\
(F/\equiv) = \neg F/\equiv
$$

$I$ denotes all formulas of theory $\Theta$. For simplicity and consistency with previous sections we write $\top$, $0$, the negations of all formulas in $\top$ is denoted by $\bot$. We obtain the Boolean algebra $< \mathcal{F}^\Gamma/\equiv_\Theta, +, \cdot, \bot, \top >$.

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 $\exists_\kappa$ 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 $S_\kappa$. $S_\kappa(F/\equiv)$ denotes the equivalence class of the formula which results from $F$ by replacing each free occurrence of $v_\kappa$ by $v_\lambda$. Here it becomes obvious why we needed an ordered sequence of variables. Extending the Boolean algebra above we obtain the quantifier algebra of formulas: $< \mathcal{F}^\Gamma/\equiv_\Theta, +, \cdot, \bot, \top, S_\kappa, \exists_\kappa >_{\kappa, \lambda < \alpha}$ associated with $\Theta$. We now formally define this algebra:

**Definition 4.3.18** By a quantifier algebra of degree $\alpha$ (QA$_\alpha$) we mean a structure

$U = < A, +, \cdot, 0, 1, S_\kappa, \exists_\kappa >_{\kappa, \lambda < \alpha}$ where $< A, +, \cdot, 0, 1 >$ is a Boolean Algebra with the unary operators $S_\kappa$ and $\exists_\kappa$ which have the following properties for all $x, y \in A$ and $\kappa, \gamma, \lambda < \alpha, \alpha \geq 2$:

1. $(q_1)$ $S_\kappa(x) = S_\kappa(x)$
2. $(q_2)$ $S_\kappa(x + y) = S_\kappa(x) + S_\kappa(y)$
3. $(q_3)$ $S_\kappa(x) = x$
4. $(q_4)$ $S_\kappa S_\lambda$
5. $(q_5)$ $\exists_\kappa(x + y) = \exists_\kappa x + \exists_\kappa y$
6. $(q_6)$ $x \leq \exists_\kappa x$
7. $(q_7)$ $S_\kappa \exists_\kappa = \exists_\kappa$
8. $(q_8)$ $\exists_\kappa S_\kappa = S_\kappa$ if $x \neq \lambda$
9. $(q_9)$ $S_\kappa \exists_\gamma = \exists_\gamma S_\kappa$ if $\gamma \neq \kappa, \lambda$

Public 84
Due to the many indexes these equations may at first look difficult but if you have a closer look you will realize 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 \(U\) is a quantifier algebra as defined above we can define a so called dimension set of a formula \(a \in A:\)

\[
\alpha x = \{ \kappa < \alpha : \forall \lambda \neq \kappa, S^\kappa_\lambda x \neq x \}
\]

Quantitatively this is the set of all indexes \(\kappa\) of variables \(v_\kappa\) which would change the valuation of formula \(x\) if substituted with another variable \(v_\lambda\). We define \(U\) to be locally finite if \(\alpha x\) is a finite set.

It can be shown that every quantifier algebra of formulas is a locally finite quantifier algebra as defined above. Accordingly, one can show that if \(U\) is a locally finite quantifier algebra then there is a theory \(\Theta\) such that \(U\) is isomorph to a quantifier algebra of formulas which could be derived from \(\Theta\).

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 formulas to \(\forall^\Gamma / \equiv_\Theta\).

Intuitivevly 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 \(e_\kappa\) which contains all formulas \(v_\kappa \equiv v_\lambda\). Finally, we define the quantifier algebra with equality.

**Definition 4.3.19** A quantifier algebra with equality is an algebra \(\langle A, +, \cdot, 0, 1, S^\kappa_\lambda, \exists_\kappa, e_\kappa >_{\kappa, \lambda < \alpha} \rangle\) such that \(\forall U < A, +, \cdot, 0, 1, S^\kappa_\lambda, \exists_\kappa >_{\kappa, \lambda < \alpha} \rangle\) is a Q\(\Lambda\alpha\) and \(e_\kappa\) are distinguished elements which satisfy

\[
\begin{align*}
(q_{10}) & \quad S^\kappa_\lambda e_\kappa = 1 \\
(q_{11}) & \quad x \cdot e_\kappa \leq S^\kappa_\lambda x
\end{align*}
\]

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 algebraization of first-order logic and show that quantifier algebra with equality is isomorph to this algebra.

Henkin, Monk and Tarski defined so called cylindric algebras in [9] as follows.

**Definition 4.3.20** By a cylindric algebra of degree \(\alpha\) we mean a system \(\langle A, +, \cdot, 0, 1, \exists_\kappa, e_\kappa >_{\kappa, \lambda < \alpha} \rangle\) such that \(\langle A, +, \cdot, 0, 1 >\) is a Boolean algebra and \(\exists_\kappa\) and \(e_\kappa\) satisfy the following conditions \(\forall x \in A\) and \(\kappa, \lambda < \alpha:\)

\[
\begin{align*}
(c_1) & \quad \exists_\kappa 0 = 0 \\
(c_2) & \quad x \leq \exists_\kappa x \\
(c_3) & \quad \exists_\kappa (x \cdot \exists_\kappa y) = \exists_\kappa x \cdot \exists_\kappa y \\
(c_4) & \quad \exists_\kappa \exists_\kappa = \exists_\kappa \exists_\kappa \\
(c_5) & \quad e_\kappa 1 = 1 \\
(c_6) & \quad e_\gamma = \exists_\kappa (e_\kappa x \cdot e_\gamma y) if \kappa \neq \gamma, \lambda \\
(c_7) & \quad \exists_\kappa (e_\kappa x \cdot \exists_\kappa (e_\kappa x \cdot x)) = 0 if \kappa \neq \lambda
\end{align*}
\]
You may already wonder why we define this algebra. If we look at the formal definition of cylindric algebras we will recognize that the substitution operation has disappeared which accounted for the $\forall$ quantifier in quantifier algebras with equation. In fact if operations $S^\kappa_\lambda$ are defined as

$$S^\kappa_\lambda x = x \text{ if } \kappa = \lambda; \quad S^\kappa_\lambda x = \exists_\kappa(x \cdot e_{\kappa \lambda}) \text{ if } \kappa \neq \lambda$$

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 $v_\kappa$ to $v_\kappa$ if $v_\kappa$ actually exists.

With this definition we can conclude that a cylindric algebra is a quantifier algebra with an equality. From [16] we additionally know that definitions $(q_1) - (q_{11})$ are actually all theorems of the theory of cylindric algebras as shown in [9].

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 [17] that we can map quantifier algebra to polyadic algebra by slightly modifying substitution and existence operators in $QA_\alpha$. By doing this we obtain a so called polyadic algebra defined by Halmos in [18].

As in section 4.3.2 we now have the evidence that we can directly map first-order logic into algebra and vice versa. Daignenault’s interpretation [19] of Krasner’s general theory on Galois Fields [20] (see also section 4.2.5) 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 Andréka, Madarász and Németi which used the relation between cylindric algebra and first-order logic to formulate Einstein’s general theory of relativity in first-order logic [21].

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, summarizing 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. Andréka et al. focus in [22] 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.

### 4.3.3 Temporal Logic

Sections 4.3.1 and 4.3.1 quantitatively introduced propositional and first-order logic. Generally speaking these logics support the reasoning based on propositions or terms and formulas. 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. Consequentially, 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 behavior.
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 shortly introduce linear temporal logic (LTL). As we will not perform a similar algebraization as for propositional and first-order logic we only give an informal definition for temporal logic and shortly 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 formulas.

1. \( \Box Q \) 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 4.3.1 as follows.

1. \( v(\Box Q) = 1 \), iff in the next time step \( v(Q) = 1 \).
2. \( v(Q \cup R) = 1 \), iff \( v(Q \land \neg R) = 1 \) until \( v(R) = 1 \).

To explain these rather abstract definitions we illustrate them in figure 4.6. 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 formulas depending on the time they are evaluated.

![Figure 4.6: Illustration of the semantics of LTL operators](image)

Clearly, the choice of the type of illustration in figure 4.6 was not arbitrary. By choosing a representation which resembles finite state machines we also wanted to emphasize the fact that Pnueli saw linear temporal logic as a tool to analyze computer programs [23].
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 $Q \lor \neg Q$ has to valuate to be true in the system. If this is true until $v(R) = 1$ the formula $Q \lor R$ valuates true for this sequence. The last formula $\bigcirc Q$ represents the simplest sequence as it only requires $Q$ to valuate to $v(Q) = 1$ in the next time step of this sequence.

With the given operators of propositional logic extended by $\bigcup$ (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. $\Diamond Q \equiv \top \bigcup Q$, with $Q \in \mathcal{P}$
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 formulas that use the operators we defined above. To clarify their meaning we again use the same illustration as in figure 4.6.

As $\Box$ depends on $\Diamond$ we start to look at the representation of $\Diamond Q$ as illustrated in figure 4.7. 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 $\bigcup$ this behavior 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 figure 4.7 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 is static characteristics, such as that the green and red light are not illuminated at the same time. Here you can 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 $\square (R \supset \Diamond G)$. 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 formulas 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 [24] Etessami 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. Consequentially, we can use the algebras developed in the last section to allow the algebraization of unary-temporal logic.

One obvious question follows: Can formulas 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 [25] and [26].

**Branching Time Logic**

So far linear temporal logics have been discussed. They get their name from the fact that they consider only behaviors which can be modelled as linear time sequences. This characteristic is nicely illustrated by figures 4.6 and 4.7. 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 [27] have been proposed. They possess 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 [28].

Although, 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 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 saw how temporal logic can extend classical propositional logic to describe properties of behavior 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 formulas are used to describe expected behavior. This behavior 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 emphasizes 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 [29]. This notion of formal verification was mainly induced by Pnueli [23, 30] and Lamport [31].

**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 [30, 31].

**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 can not do. Formulas of (temporal) logic can be used to specify these requirements.

### 4.4 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 [32]. A more detailed discussion of these physics concepts and their relationship to self-organisation can
be found in [3]. 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) [33], 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 4.3.2 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 [34]. 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 4.3.3 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.
Bibliography


Chapter 5

Evolutionary Computing and Artificial Embryogeny

L. Yamamoto, D. Miorandi

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 is grown 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 online, 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 the aforementioned techniques to the BIONETS environment.

5.1 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, autonomic 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, online 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 online 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. 5.2 we position our context within the classification adopted from [41]. In Sec. 5.3 we review the state-of-the-art in evolutionary computing with focus on online and dynamic environments. In Sec. 5.4 we present the two main research lines inspired by embryology: embryonics and artificial embryogenies. Sec. 5.5 presents a critical discussion on the possible combination of the aforementioned approaches. Sec. 5.6 concludes the chapter pointing out possible applications to BIONETS.

5.2 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 (Evolvable Hardware) or in software (Evolutionary Computation). The latter will be described in Section 5.3.

- **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 5.4.

The POE classification is represented in Fig. 5.1, 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,
Figure 5.1: The POE classification and some of the phylogenetic and ontogenetic approaches that will be treated in this paper.

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 online 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 online environment. The potential of such combined PO-Plane approaches will be discussed in Section 5.5.

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.

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 online 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.

5.3.1 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.

5.3.2 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 online evolution. Indirect encodings will be briefly discussed in Section 5.3.6.
Most GP approaches can be placed along the “P” axis in the POE framework (Fig. 5.1). 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 5.4.2.

5.3.3 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 subpopulations.
- 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 subpopulations 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.

5.3.4 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.

5.3.5 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.

### 5.3.6 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.

### 5.3.7 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.

5.3.8 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 once. 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 online 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 signaling 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 online problems, to the best of our knowledge [2] has not been applied in this context.

5.4 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 behaviors. 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.

5.4.1 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 neighboring cells, which functionalities (genes) need to be expressed.

- The system possesses self-organizing properties. Each cell monitors its neighborhood 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 neighboring 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. 5.2, we can position it as a pure ontogenetic approach.

5.4.2 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/behavior 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 5.3.5), 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.

5.5 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.
5.6 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 online 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.
Bibliography


6.1 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 needs 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 guarantee 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 appeared in the literature, and provide a comprehensive survey of the related works. In Sec. 6.2, we analyze 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. 6.3 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. 6.4, we describe which are 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. 6.5 concludes the survey by drawing the conclusions of this work, and by pointing out the most promising research directions that are currently being investigated.

### 6.2 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 media 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 (6.2.1, 6.2.2 and 6.2.3) 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 6.2.1 we show what kind of information can be used to optimize broadcasting.

### 6.2.1 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 needed information 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 an 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 neighbors, 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 try 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 behavior 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 good performing heuristics, that can be found by learning algorithms or genetic algorithms (or even by a combination of both).
Neighbor 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 neighbors of the sender. Some of the algorithms collect knowledge about their immediate neighbors 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 nodes speed, 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 decision. 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 neighbor information (section 6.2.1) algorithms do, but they collect the location of the neighbors too. There are also algorithms that need to know only the distance to their neighbors [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 neighbor 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.

6.2.2 Categorization by strategy

In section 6.2.1 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 behavior that do not come from the randomness of the environment. The benefit of this can be the elimination of coupling between the decisions of neighbors, 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 behavior may destroy some information for the neighboring nodes, as the behavior of the algorithm is no longer deterministic. Usually the behavior is adaptable to different situations by adjusting the probabilities of different decisions. Many of the stochastic algorithms are heuristic based. Some of the algorithms have some Media Access Control that introduces non-deterministic behavior. 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 6.2.1) and behave always the same when the environment is the same. One of the simplest examples is the Counter Based Method already discussed in subsection 6.2.1. Among the deterministic algorithms, the ones 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 neighbors should be forwarding nodes.

The self-pruning algorithms collect neighborhood 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 neighborhood 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 neighbor 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 neighbors to be forwarding nodes for its 2-hop neighbors. The list of the designated forward nodes is usually sent with the broadcast packet [68]. The goal for neighbor 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 neighbor designation algorithms as special cases.

6.2.3 Categorization by media access

In every problem which have to be solved over 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 unresponsible 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 neighbors). We should note that the use of RAD creates non-deterministic behavior, 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.

### 6.2.4 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 neighbor 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 neighbors of a node rebroadcast, thus minimizing overlap of transmission ranges. It uses 1-hop neighbor information and records signal levels from the neighbor nodes. The authors propose two variants called DAD-NUM and DAD-PER. DAD-NUM chooses a signal strength $S_{\text{thresh}}$ so that there are $k$ number of neighbors that have transmitted with a signal strength lower than $S_{\text{thresh}}$. On arrival of a new packet, the node checks if the signal strength is greater or $S_{\text{thresh}}$ 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 neighbor information and the last sender ID in the broadcast packet. When a node $v$ receives a broadcast packet from a node $u$, $N(u)$ from the set of its own neighbors $N(v)$. 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 $(d_v/d_{\text{max}}) \cdot T_{\text{max}}$ formula, where $d_v = |N(v)|$ and $d_{\text{max}}$ is the degree of the node with the largest degree in $N(v)$, and $T_{\text{max}}$ 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 neighbors.

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 neighbors. When one of them starts broadcasting, SBA removes the neighbors of the broadcasting nodes from its own 1-hop neighbor set. This process is demonstrated on figure 6.1. The length of the RAD depends on the number of the immediate (1-hop) neighbors, so nodes with more neighbors...
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 neighbors 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 neighbor information. Every time a HELLO message is heard, MMSBA updates the neighbors 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 neighbor 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 neighbors 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 neighbors.

**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 neighbors 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 back-off 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 neighbors of \( u \), from its own neighbor set \( N(v) \).

Multipoint relaying [64] is a neighbor designation protocol. The designated nodes that relay the messages are called Multipoint Relays (MPR). The nodes send HELLO messages to discover their 2-hop neighborhood and they try to choose as MPR the node that is able to reach most nodes among the 2-hop neighbors.
The algorithm first chooses the nodes from its 2-hop neighbors that are reachable by only one node from the 1-hop neighbors, and assigns MPR status to these 1-hop neighbors. From the remaining set of 1-hop neighbors it chooses the one that covers most of the uncovered 2-hop neighbors. This step is repeated until all of the 2-hop neighbors 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 neighbors (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 neighbors 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 neighbor 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 neighborhood 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 neighbor and 2-hop routing information is used, and the self-pruning made according to the static version of Coverage Condition I: Node $v$ has a non-forwarding status if for any two neighbors $u$ 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 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 neighbor 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 neighbors. It uses 2-hop neighbor and 1-hop routing information.

---

1 Coverage Condition II is a computationally less expensive approximation of Condition I for very simple devices.
A quite different approach from the algorithms discussed so far is the IOBIO algorithm \cite{63}. It is a variation of the SPIN \cite{27} dissemination protocol. It uses a simple 3-stage handshake to discover neighbors that are interested in one of the carried messages. The goal of the protocol is to reduce the unnecessary load of neighboring 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. Neighbor 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 6.2. 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.

### 6.2.5 General comparison and discussion

Table 6.1 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 neighbor 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 neighbor based schemes do not need any additional support. In dense scenarios neighbor 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 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.

### 6.3 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) \cite{1},\cite{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 \cite{54}, deep space networks...
Table 6.1: Comparison of broadcast algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Multi-message</th>
<th>Forwarding Decision</th>
<th>Control Messages</th>
<th>Special Hardware</th>
<th>Strategy</th>
<th>Sensitivity on Mobility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter Based [53]</td>
<td>no</td>
<td>self</td>
<td>none</td>
<td>none</td>
<td>Simple heuristic</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>Gossiping [53]</td>
<td>no</td>
<td>self</td>
<td>none</td>
<td>none</td>
<td>Simple heuristic / Stochastic</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>Hypergossip [37]</td>
<td>yes</td>
<td>self</td>
<td>LBR and BR</td>
<td>none</td>
<td>Stochastic</td>
<td>Speed</td>
</tr>
<tr>
<td>OFP [62]</td>
<td>no</td>
<td>self</td>
<td>HELLO with location</td>
<td>circular radio range; GPS</td>
<td>Geometry based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>DAD [39]</td>
<td>no</td>
<td>self</td>
<td>HELLO</td>
<td>signal strength measurement</td>
<td>Simple heuristic / Location based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>SBA [45]</td>
<td>no</td>
<td>self</td>
<td>2-hop HELLO</td>
<td>none</td>
<td>Neighbor based</td>
<td>High speeds and Connectedness</td>
</tr>
<tr>
<td>MMSBA [56]</td>
<td>yes</td>
<td>self</td>
<td>2-hop HELLO with BR</td>
<td>none</td>
<td>Neighbor based</td>
<td>Speed</td>
</tr>
<tr>
<td>Wu and Li’s algorithm [42]</td>
<td>no</td>
<td>self</td>
<td>2-hop HELLO</td>
<td>none</td>
<td>Neighbor based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>Stojmenovic [18, 43]</td>
<td>no</td>
<td>self</td>
<td>2-hop HELLO</td>
<td>GPS</td>
<td>Neighbor based / Location based</td>
<td>High speeds and Connectedness</td>
</tr>
<tr>
<td>MPR [64]</td>
<td>no</td>
<td>designated</td>
<td>2-hop HELLO</td>
<td>none</td>
<td>Neighbor based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>AHBP [46]</td>
<td>no</td>
<td>designated</td>
<td>2-hop HELLO</td>
<td>none</td>
<td>Neighbor based</td>
<td>High speeds and Connectedness</td>
</tr>
<tr>
<td>Generic [68]</td>
<td>no</td>
<td>self</td>
<td>k-hop HELLO</td>
<td>may use for prioritization</td>
<td>Neighbor based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>LENWB [68]</td>
<td>no</td>
<td>self</td>
<td>2-hop HELLO</td>
<td>none</td>
<td>Neighbor based</td>
<td>Speed and Connectedness</td>
</tr>
<tr>
<td>MIOBIO [63]</td>
<td>yes</td>
<td>handshake</td>
<td>BR and REQ</td>
<td>none</td>
<td>Handshake based</td>
<td>Speed</td>
</tr>
</tbody>
</table>

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 afterward 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 opportunities 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.

### 6.3.1 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 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.

### 6.3.2 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

---

**Footnote:** Two nodes are in contact if they are within transmission range of one another.

---

Public 126
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 modeled 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 behaviors, 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.

6.3.3 Predicted-contact based routing approaches

Predicted routing techniques attempt to take advantage of certain knowledge concerning the mobility patterns or some repeating behavioral 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 behaviors 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 neighbor 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 neighbors \( W_X(t) \) 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 behavioral 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 increases 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 neighbor 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 neighbors, it requests the hierarchy level of all of its neighbors. Collected data is then sent to the neighbor 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 forwards of messages to each other, and hence they will reduce their corresponding delivery predictability entries.

6.3.4 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
November 5, 2008

D2.1.1 Survey on paradigms

BIONETS//wp2.1/0.1

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 analyti-
cal 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 modeling 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 au-
thors 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 6.3.4 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 \( k \approx \frac{m}{l} < n \) 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 \( a \ xor \ c \) 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.

### 6.3.5 General comparison and discussion

Table 6.2 compares the various proposals that have been addressed in Section 6.3 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
Table 6.2: Summary of the routing approaches in DTNs and their main properties.

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Contact opportunities</th>
<th>Metric to optimize</th>
<th>Mobility pattern of network nodes</th>
<th>Mobility pattern of special nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. protocol [31]</td>
<td>Scheduled</td>
<td>Delay</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>MULE protocol [32]</td>
<td>Controlled</td>
<td>Power usage, Buffer overhead</td>
<td>Stationary</td>
<td>Random</td>
</tr>
<tr>
<td>MF protocol [70]</td>
<td>Controlled</td>
<td>Delivery rate, Power usage</td>
<td>Stationary</td>
<td>Predetermined paths</td>
</tr>
<tr>
<td>Extended MF protocol [71]</td>
<td>Controlled</td>
<td>Delivery rate, Power usage</td>
<td>Random, Stationary</td>
<td>Predetermined, Dynamic paths</td>
</tr>
<tr>
<td>MV protocol [14]</td>
<td>Controlled</td>
<td>Delivery rate, Delay</td>
<td>Meeting and Visit dependant</td>
<td>Metric dependant paths</td>
</tr>
<tr>
<td>Island hopping protocol [55]</td>
<td>Predicted</td>
<td>Delay, Transmission overhead</td>
<td>Heterogeneous mobility</td>
<td>–</td>
</tr>
<tr>
<td>MobySpace protocol [41]</td>
<td>Predicted</td>
<td>Delivery rate, Power usage</td>
<td>Location dependant</td>
<td>–</td>
</tr>
<tr>
<td>DLE protocol [19]</td>
<td>Predicted</td>
<td>Buffer usage</td>
<td>Heterogeneous mobility</td>
<td>–</td>
</tr>
<tr>
<td>ZebraNet [35]</td>
<td>Predicted</td>
<td>Delivery rate, Power, Storage</td>
<td>Heterogeneous mobility</td>
<td>–</td>
</tr>
<tr>
<td>MEED protocol [34]</td>
<td>Predicted</td>
<td>Delay, Transmission overhead</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>PROPHET [44]</td>
<td>Predicted</td>
<td>Delivery rate, Power usage</td>
<td>Heterogeneous mobility</td>
<td>–</td>
</tr>
<tr>
<td>Epidemic protocol [61]</td>
<td>Opportunistic</td>
<td>Delivery ratio, Delay</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>Two-hop protocol [24]</td>
<td>Opportunistic</td>
<td>Network capacity</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>MTR protocol [23]</td>
<td>Opportunistic</td>
<td>Delay, Transmission overhead</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>SWIM protocol [57]</td>
<td>Opportunistic</td>
<td>Delivery rate, Delay</td>
<td>Random, Stationary</td>
<td>–</td>
</tr>
<tr>
<td>Spary and Wait [59]</td>
<td>Opportunistic</td>
<td>Delivery rate, Power usage</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>Erasure coding [65]</td>
<td>Opportunistic</td>
<td>Delivery rate</td>
<td>Random</td>
<td>–</td>
</tr>
<tr>
<td>Network coding [66]</td>
<td>Opportunistic</td>
<td>Delivery rate</td>
<td>Random</td>
<td>–</td>
</tr>
</tbody>
</table>

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 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.

6.4 Modelling approaches

In the absence of predictable mobility and network topology, the notion of “route” for a message to follow loosens 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 traveling 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 modeling 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 modeling 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 6.3. 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 6.3.

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 parameterized 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 modeling approaches of epidemics can not be applied. The most important limitation is that mobility can not be modeled by exponential intermeeting times because of the significant probability of having a node already in range. Also meetings can not be assumed pairwise anymore 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.

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 6.3 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.

6.5 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.
Bibliography


Part II

Paradigms from Physics
Chapter 7

Scale-free networks

J. Riihijarvi, P. Mahonen, F. Oldewurtel (RWTH)

Abstract. We discuss the role 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 characterizing spatial structure 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.

7.1 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 proba-
bilistic 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 hotspot networks. Techniques for constructing probabilistic spatial network models are discussed, and finally the conclusions are drawn.

7.2 Scale-free relational network models

Fixed networks are usually modelled using various types of random graph models, that is, ensembles ($\mathcal{G}, G, \mathbb{P}_{G}$) of graphs in some large graph space $\mathcal{G}$. The “modelling” comes in via the definition of the probability measure $\mathbb{P}_{G}$ of a given graph to be realised. Until recently $\mathbb{P}_{G}$ 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

$$\mathbb{P}\{d(v) = k | v \in V(G_{n,p})\} = \binom{n-1}{k} p^k (1-p)^{n-1-k}, \quad (7.1)$$

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]. 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 $\Pi(i) = d(i)/\sum_j d(j)$. 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 7.1.

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

$$\frac{\partial d_i(t)}{\partial t} = m\Pi(i) = \frac{d_i}{t_i}.$$ (7.2)

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 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 $\ln n / \ln \ln n$.

\[^1\text{It should be pointed out that Price gave a very similar model much earlier in [36], but this has remained practically unknown for the applied graph theory community until recently.}\]
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 structure of the present-day networks as an outcome of a large collection of local optimization procedures. Nevertheless, their applications have also been mainly confined to toy-models, leaving ample room for further work on these matters.

### 7.3 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 $N = \sum_i \delta_{x_i}$, where $\delta_x$ 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 $\mu$. 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
\[ dP = \nu^2 (1 + \xi(r)) dA_1 dA_2 \] 
(7.3)
of finding one point in each of the two area elements \( dA_1 \) and \( dA_2 \) with \( \nu \) 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 \( \xi \) 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 power-
ful 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 correla-
tions. The definition analogous to (7.3) is given by the joint probability density
\[ dP = \nu^n (1 + \xi^{(n)}(r_{1,2}, r_{1,3}, \ldots, r_{n-1,n})) \prod_{i=1}^{n} dA_i, \]
(7.4)
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
\[
\begin{align*}
dP &= \nu^3 (1 + \xi^{(3)}(r_{1,2}, r_{1,3}, r_{2,3})) dA_1 dA_2 dA_3 \\
&= \nu^3 (1 + \xi(r_{1,2}) + \xi(r_{1,3}) + \xi(r_{2,3}) + \zeta(r_{1,2}, r_{1,3}, r_{2,3})) dA_1 dA_2 dA_3,
\end{align*}
\]
(7.4)
where \( \zeta \) 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
\[ \Phi_r(x, y) \equiv |r \leq d(x, y) \leq r + \Delta|, \]
(7.5)
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
\[ DD(r) = \sum_{x \in D} \sum_{y \in D \setminus \{x\}} \Phi_r(x, y) / (n(n - 1)). \]
(7.6)
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
\[ DR(r) = \sum_{x \in D} \sum_{y \in R} \Phi_r(x, y) / (nm) \]
(7.7)
and
\[ RR(r) = \sum_{x \in R} \sum_{y \in R \setminus \{x\}} \Phi_r(x, y) / (m(m - 1)), \]
(7.8)
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

$$\hat{\xi}_N \equiv \frac{DD}{RR} - 1, \quad \hat{\xi}_{DP} \equiv \frac{DD}{DR} - 1 \quad \text{and} \quad \hat{\xi}_{He} \equiv \frac{DD - DR}{RR}. \quad (7.9)$$

More recently new estimators have been suggested by Hamilton [22] and Landy and Szalay [30], which are given by

$$\hat{\xi}_{Ha} \equiv \frac{DDRR}{DR^2} - 1 \quad \text{and} \quad \hat{\xi}_{LS} \equiv \frac{DD - 2DR + RR}{RR}. \quad (7.10)$$

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 7.3 for an illustration. The corresponding estimates of the two and (reduced) three-point correlation functions are shown in figure 7.3. 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

---

Figure 7.2: The WLAN access point locations featuring scale-free spatial structure. (Adapted from [37].)
Figure 7.3: 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].)

distributions with scale-free characteristics in terms of correlations.

7.4 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 $\xi(r)$ 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 realizations 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 realizations 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 interpoint 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 \( \mathcal{A} \).

Suppose further that the law of \( N \) is given by a density function \( f : \mathbb{R}^{nd} \rightarrow [0, \infty) \), that is,

\[
P \{ X_1, \ldots, X_n \in B \} = \int_B f(x_1, \ldots, x_n) \, dx_1 \cdots dx_n. \tag{7.11}
\]

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* \( H(X_1, \ldots, X_n) \). 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

\[
S = -\int_{E^n} f(x_1, \ldots, x_n) \ln f(x_1, \ldots, x_n) \, dx_1 \cdots dx_n \tag{7.12}
\]

under the constraint of constant \( H \) in expectation. This yields the density

\[
f(x_1, \ldots, x_n) = \exp(-H(x_1, \ldots, x_n))/Z, \tag{7.13}
\]

where the partition function \( Z \) is given by

\[
Z = \int_{E^n} \exp(-H(x_1, \ldots, x_n)) \, dx_1 \cdots dx_n. \tag{7.14}
\]

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 \( H(X_1, \ldots, X_n) \). A common choice considered in applications has been the *pairwise interaction model* defined by

\[
H(X_1, \ldots, X_n) = a_0 + \sum_{i=1}^{n} \psi(X_i) + \sum_{1 \leq i < j \leq n} \phi(X_i, X_j) \tag{7.15}
\]

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.

### 7.5 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 re-
search avenues in the graph theory front. Comparatively less work has been done in characterizing 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 realizations 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.
Bibliography


Chapter 8

Random matrix theory and free probability

Merouane Debbah

8.1 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 uplink 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 optimize 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, centered on the applications to Information Theory, 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 $v_1 v_1^H + \cdots + v_n v_n^H$ where $v_i$ are i.i.d. Gaussian vectors. His results are among the few available on finite dimensional matrices. The typical question is to characterize 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-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-organizing 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 *noncommutative 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 analyzed. 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 spectrums) on the network. In this paper, we give an explicit relation between the spectrums of random matrices \((M + N)(M + N)^\ast\), \(MM^\ast\) and \(NN^\ast\), where \(M, N\) 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 \(M\) or \(N\) is gaussian.

### 8.2 Definitions and Notations

We consider the general case of two independent real rectangular random matrices \(M, N\), 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* \(MM^\ast\) and \(NN^\ast\) (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 Gram\(^1\) assumption was mainly due to the fact that the eigenvalues (which are real and positive) are easier to characterize. From an engineering perspective, for a random network modeled by a matrix \(M\), the eigenvalues of \(MM^*\) contain all the sufficient information to characterize 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 \((M + N)(M + N)^*\) from the limit eigenvalue distributions of \(MM^*\) and \(NN^*\). The underlying operation on probability measures is called the rectangular free convolution with ratio \(\lambda\) and denoted by \(\boxplus\lambda\) in the literature ([24, 25, 26]). Our machinery will also allow the inverse operation, called rectangular deconvolution with ratio \(\lambda\): the derivation of the eigenvalue distribution of \(MM^*\) from the ones of \((M + N)(M + N)^*\) and \(NN^*\).

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 \(A, B\) independent large square hermitian (or symmetric) random matrices (under some general hypothesis that will be specified):

- Derive the eigenvalue distribution of \(A + B\) from the ones of \(A\) and \(B\).
- Derive the eigenvalue distribution of \(AB\) or of \(A^\dagger B A^\dagger\) from the ones of \(A\) and \(B\).

The previous operations are known in the literature as free convolutions ([27]), and denoted respectively by \(\boxplus, \boxminus\).

We will also see how one can:

- Deduce the eigenvalue distribution of \(A\) from the ones of \(A + B\) and \(B\).
- Deduce the eigenvalue distribution of \(A\) from the ones of \(AB\) or of \(A^\dagger B A^\dagger\) and \(B\).

These last operations are called free deconvolutions ([21]) and denoted respectively by \(\boxminus, \boxminus\).

### 8.3 Information in random networks

In wireless intelligent random networks, devices are autonomous and should take optimal decisions based on their sensing capabilities. Of particularly 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:

\[
H = \log(\pi e) + \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(S(f)) df,
\]

\(^1\)For a matrix \(M\), \(MM^*\) is called the Gram matrix associated to \(M\)
where $S$ is the spectral density of the process. Hence, if one knows the autocorrelation of the process, one has therefore a full characterization of the information contained in the process. Moreover, as 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 autocorrelation function for Gaussian processes. In the discrete case when considering a random Gaussian vector $\mathbf{x}$ of size $n$, the entropy rate per dimension (or differential entropy) is given by:

$$
H = \log(\pi e) + \frac{1}{n} \log \det(R)
$$

where $R = \mathbb{E}(\mathbf{x}\mathbf{x}^*)$ is the covariance and $\lambda_i$ the associated eigenvalues. The covariance (and more precisely its eigenvalues) carries therefore all the information of Gaussian networks. The Gaussianity of these networks is due to the fact that the noise, the channel and the signaling 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 $\mathbf{x}_i, i = 1, ..., p$, the covariance $R$ is usually estimated by:

$$
\hat{R} = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i \mathbf{x}_i^*
$$

$$
= \mathbf{R} + \mathbf{SS}^* \mathbf{R}^T
$$

Here, $\mathbf{S} = [s_1, ..., s_p]$ is an $n \times p$ i.i.d zero mean Gaussian vector of variance $\frac{1}{p}$. It turns 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 $p$ number 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 asymptotics "kick in" 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 (8.3) and express $\mathbf{R}$ with respect to $\hat{R}$. As we will show later on, this is not possible, however, one can compute the eigenvalues of $\mathbf{R}$ knowing only the the eigenvalues of $\hat{R}$ (as the limiting eigenvalues of $\mathbf{SS}^*$ 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 $\mathbf{SS}^*$), the eigenvector structure does not matter. Otherwise, it would have not been possible. The invariance assumption "frees" in some sense one matrix from the other by "disconnecting" their eigenspaces.
8.4 Eigenvalue distribution, joint eigenvalue distribution and moments

8.4.1 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 have 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 characterize 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)^2. Note that \( C_n \) is a random variable (due to the fact that the eigenvalues \( \lambda_i \) are random) and can be expressed as:

\[ C_n = \int f(\lambda) \frac{1}{n} \delta(\lambda - \lambda_i) d\lambda \quad (8.4) \]

\[ = \int f(\lambda) d\rho_n(\lambda) \quad (8.5) \]

We call here \( \rho_n \) the eigenvalue distribution which turns out to be nothing else than a projection of the vector of eigenvalues \( [\lambda_1, \ldots, \lambda_n] \) into a single quantity. It is therefore less explicit than the joint eigenvalue distribution but contains all what is required for our problem. \( \rho_n \) is also a random variable but as the dimension of the system grows, the behavior of the eigenvalue distribution becomes deterministic in many cases. We will denote it \( \rho \). Of course, other projections may also be of interest in other applications such as the maximum of the vector \( [\lambda_1, \ldots, \lambda_n] \).

8.4.2 Eigenvalue distribution and moments

Let us consider a probability measure \( \rho \) 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 \( \{m_k, k \geq 0\} \) does not uniquely determine the associated probability distribution. A trivial sufficient condition is the existence of the moment generating function\(^3\). In any case, for computing the eigenvalue distribution, one needs to determine the moments of all orders. 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).

---

^2 In general, the function \( f \) should have other constraints (bounded) but for clarity reasons, we do not go into more details.

^3 A more sophisticated one is the Carleman condition, which states that the sequence characterizes a distribution if the following holds:

\[ \sum_{j=1}^{\infty} m_{2j}^{\frac{1}{j}} = \infty \]
In this case, the moments are related to the eigenvalues by the following relations:

\[ m_k(p) := \frac{1}{L} \sum_{i=1}^{L} \lambda_i^k. \]  

(8.6)

As detailed in [31, 19, 32], one needs only to compute \( L \) moments to retrieve the eigenvalues in equation (8.6). This simplifies drastically the problems and favors a moment approach to the free deconvolution framework rather than deriving the explicit spectrum.

The Newton-Girard Formulas [31] can be used to retrieve the eigenvalues from the moments. These formulas state a relationship between the elementary symmetric polynomials

\[ \Pi_j(\lambda_1, \ldots, \lambda_L) = \sum_{i_1 < \cdots < i_j \leq L} \lambda_{i_1} \cdots \lambda_{i_j}, \]  

(8.7)

and

\[ S_p(\lambda_1, \ldots, \lambda_L) = \sum_{i=1}^{L} \lambda_i^p = L \times m_p(p) \]

through the recurrence relation

\[ (-1)^m m \Pi_m(\lambda_1, \ldots, \lambda_L) + \sum_{k=1}^{m} (-1)^{k+m} S_k(\lambda_1, \ldots, \lambda_L) \Pi_{m-k}(\lambda_1, \ldots, \lambda_L) = 0. \]  

(8.8)

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: \((-1)^k \Pi_k(\lambda_1, \ldots, \lambda_L)\). As \( m_p(p) \) (we will show later on how these quantities can be computed) are known for \( 1 \leq p \leq L \), (8.8) can be used repeatedly to compute \( \Pi_m(\lambda_1, \ldots, \lambda_L), 1 \leq m \leq L \).

### 8.4.3 Information plus Noise model

Example (8.3) is unfortunately rarely encountered in practice in wireless communications. The signal of interest \( s_i \) is usually distorted by a medium, given by \( m_i = f(s_i) \) where \( f \) is any function. Moreover, the received signal \( y_i \) is altered by some additive noise \( n_i \) (not necessarily Gaussian) but in many respects unitarily invariant (due to the fact that all the dimensions have the same importance). In this case, the model is known as the Information plus Noise model:

\[ y_i = m_i + n_i, \]

which can be rewritten in the following matrix form by stacking all the observations:

\[ Y = M + N. \]  

(8.9)

The main challenge here is to infer on the eigenvalues of \( MM^* \) when \( YY^* \) and the eigenvalues of \( NN^* \) are known (indeed, the noise can be measured by opening the receptor before the arrival of the signal...
Once the spectrum of $\mathbf{MM}^*$ 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 Stieltjes transform [23] and other related methods for specific [33, 34, 35] models (typically, the case where $\mathbf{N}$ is i.i.d Gaussian). In this work, we will show however that the cumulants/moment approach can be a very efficient tool and provide implementable algorithms for a broader class of models.

### 8.5 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 Schrödinger equation:

$$\mathbf{H}\phi_i = E_i\phi_i$$

where

- $\phi_i$ is the wave function
- $E_i$ is the energy level
- $\mathbf{H}$ is the hamiltonian

Hence, the energy levels of the operator $\mathbf{H}$ is nothing else than the eigenvalues of the matrix representation of that operator. For a specific nuclei, finding the exact eigenvalues is a very complicated problem as the number of interacting particles increase. The genuine idea of Wigner was to replace the exact matrix by a 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 irrespective of the exact realization 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.

### 8.6 Theoretical background

In the following, upper case and lower case boldface symbols will be used for matrices and column vectors, respectively. $(.)^T$ will denote the transpose operator, $(.)^*$ conjugation and $(.)^H = (.)^T)^*$ hermitian transpose. $E$ denotes the expectation operator.
**Definition 8.6.1** \( v = [v_1, \ldots, v_N] \) be a vector. Its empirical distribution is the function \( F_{v_N}^N : \mathbb{R} \to [0, 1] \) defined by:

\[
F_{v_N}^N(x) = \frac{1}{N} \# \{ v_i \leq x \mid i = 1 \ldots N \}.
\]

\# corresponds to the cardinality operator. In other words, \( F_{v_N}^N(x) \) is the fraction of elements of \( v \) that are inferior or equal to \( x \). In particular, if \( v \) is the vector of eigenvalues of a matrix \( V \), \( F_{v_N}^N \) is called the empirical eigenvalue distribution of \( V \).

A Wigner matrix is an \( N \times N \) symmetric matrix \( H \) with diagonal entries zero and upper-triangle entries i.i.d. zero mean and variance 1. As \( N \to \infty \), the empirical eigenvalue distribution of \( \frac{1}{\sqrt{N}} H \) converges to the semicircle law:

\[
f(\lambda) = \begin{cases} 
\frac{1}{\pi} \sqrt{4 - \lambda^2} & \text{if } |\lambda| \leq 2 \\
0 & \text{if } |\lambda| \geq 2
\end{cases}
\]

The semicircle law is plotted in Fig. 8.1, as well as the plot obtained by tracing the histogram of the eigenvalues of a single realization of a \( 512 \times 512 \) 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 nonzero entries are not bounded, the distribution of the eigenvalues has a bounded support.

Wigner matrices have quite a constrained form, but it also possible to obtain results for a non-symmetric matrix. If \( H \) is an \( N \times N \) matrix with entries i.i.d. zero mean and variance 1, then the eigenvalues of \( \frac{1}{\sqrt{N}} H \) 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. 8.2, as well as the plot of the eigenvalues of a single realization of a \( 512 \times 512 \) 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 \( H \) be an \( N \times K \) matrix, with i.i.d. zero-mean complex entries with variance \( \frac{1}{N} \) and fourth moments \( O \left( \frac{1}{N^2} \right) \). As \( K, N \to \infty \),
with $\frac{K}{N} \to \alpha$, the empirical eigenvalue distribution of $\mathbf{H}^T \mathbf{H}$ 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 $b = (1 + \sqrt{\alpha})^2$.

The Marchenko-Pastur law is plotted in Fig. 8.3. The asymptotic analysis has an averaging effect: the limit distribution depends only on $\alpha$, 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.

### 8.7 Moments approach

#### 8.7.1 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:

$$m_1^N = \frac{1}{N} \text{trace} (\mathbf{H}) = \frac{1}{N} \sum_{i=1}^{N} \lambda_i = \int \lambda dF_N(\lambda)$$

$$m_2^N = \frac{1}{N} \text{trace} (\mathbf{H})^2 = \int \lambda^2 dF_N(\lambda)$$

... = ...

$$m_k^N = \frac{1}{N} \text{trace} (\mathbf{H})^k = \int \lambda^k dF_N(\lambda)$$

Figure 8.2: Full circle law and simulation for a $512 \times 512$ matrix.
Figure 8.3: Marchenko-Pastur density function for $\alpha = 1, 0.5, 0.2$. 
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:

$$\lim_{N \to \infty} \frac{1}{N} \text{Trace}(H^{2k}) = \int_{-2}^{2} x^{2k} f(x) \, dx = \frac{1}{k+1} C_{k}^{2k}$$

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 provided by:

$$f(x) = \frac{1}{2\pi} \sqrt{4-x^2}$$

with $|x| \leq 2$. Once can verify it directly by calculus based on recursion:

$$\alpha_{2k} = \frac{1}{\pi} \int_{-2}^{2} x^{2k} \sqrt{4-x^2} \, dx$$

$$= - \frac{1}{2\pi} \int_{-2}^{2} \frac{x}{\sqrt{4-x^2}} \times x^{2k-1}(4-x^2) \, dx$$

$$= \frac{1}{2\pi} \int_{-2}^{2} \sqrt{4-x^2} (x^{2k-1}(4-x^2))' \, dx$$

$$= 4(2k-1)\alpha_{2k-2} - (2k+1)\alpha_{2k}$$

In this way, the recursion is obtained:

$$\alpha_{2k} = \frac{2(2k-1)}{k+1} \alpha_{2k-2}$$

### 8.7.2 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 $HH^H$ where $H$ is $N \times K$ i.i.d Gaussian variance $\frac{1}{N}$ with $K_N = \alpha$. In this case, in the same manner, the moments of this distribution are given by:

$$m_1^N = \frac{1}{N} \text{trace}(HH^H) = \frac{1}{N} \sum_{i=1}^{N} \lambda_i \to 1$$

$$m_2^N = \frac{1}{N} \text{trace}(HH^H)^2 = \frac{1}{N} \sum_{i=1}^{N} \lambda_i^2 \to 1 + \alpha$$

$$m_3^N = \frac{1}{N} \text{trace}(HH^H)^3 = \frac{1}{N} \sum_{i=1}^{N} \lambda_i^3 \to \alpha^2 + 3\alpha + 1$$

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

\[
\begin{bmatrix}
N & H \\
H & -K
\end{bmatrix}
\]

when \(N \to \infty\), \(K/N \to \alpha\) 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 \(O\left(\frac{1}{N^2}\right)\). Therefore, the distance of the matrix to the identity matrix (in the Froebenius norm sense) is not zero.

8.8 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 noncommutative probability space. This can be defined by a pair \((A, \phi)\), where \(A\) is a unital \(*\)-algebra with unit \(I\), and \(\phi\) is a normalized (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, \(\phi\) will be the normalized trace \(tr_n\), defined by

\[
(tr_n(a)) = \frac{1}{n} Tr(a) = \frac{1}{n} \sum_{i=1}^{n} a_{ii},
\]

while for random matrices, \(\phi\) will be the linear functional \(\tau_n\) 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 \(I_n\). The noncommutative probability spaces considered will all be tracial, i.e. \(\phi\) satisfies the trace property \(\phi(ab) = \phi(ba)\). The analogy to independence is called freeness:

**Definition 8.8.1** A family of unital \(*\)-subalgebras \((A_i)_{i \in I}\) will be called a free family if

\[
\begin{cases}
  a_j \in A_{i_j} \\
  i_1 \neq i_2, i_2 \neq i_3, \ldots, i_{n-1} \neq i_n \\
  \phi(a_1) = \phi(a_2) = \cdots = \phi(a_n) = 0
\end{cases}
\]

\(\Rightarrow \phi(a_1 \cdots a_n) = 0\). (8.10)

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 \(\phi\) is a trace and \(i_1 = i_n\), we can rearrange the terms so that two consecutive terms in (8.10) 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 (8.10) is satisfied for all indices where the
We will say that a sequence of random variables \( a_n \) is called asymptotically free if \( \phi(a_n) \) converges in distribution with the moments of \( A \) as \( n \to \infty \). For instance, consider random matrices \( A_n \) asymptotically free. The limit distribution of the first term is zero by assumption, since \( \phi(b_1) = 0 \), \( b_1 \in A_n \) and \( n \to \infty \). The second term \( \phi(a_n) \phi(a_2 \cdots a_{n-1}) \) contributes with zero when \( i_2 = n-1 \) by assumption. If \( i_2 = n-1 \), we use the same splitting as in (8.11) 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 \( i_3 = n-2 \). Continuing in this way, we will 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 \( i_{n/2} \neq n/2+1 \), and the second is 0 by assumption.

**Definition 8.8.2** We will say that a sequence of random variables \( a_n \) in probability spaces \( (A, \phi) \) converge in distribution if, for any \( m_1, \ldots, m_r \in \mathbb{Z}, k_1, \ldots, k_r \in \{1, 2, \ldots\} \), we have that the limit \( \phi_n(a_{m_1} a_{m_2} \cdots a_{m_r}) \) exists as \( n \to \infty \). If these limits can be written as \( \phi(a_1 \cdots a_{k_1} \cdots a_{k_2} a_{m_r}) \) for some noncommutative probability space \( (A, \phi) \) and free random variables \( a_1, a_2, \ldots \in (A, \phi) \), we will say that the \( a_n \) 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 \( \frac{1}{\sqrt{n}} A_{n1}, \frac{1}{\sqrt{n}} A_{n2}, \ldots \), where the \( A_{ni} \) 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 \( \frac{1}{\sqrt{n}} A_{ni} \) are asymptotically free. The limit distribution of the \( \frac{1}{\sqrt{n}} A_{ni} \) in this case is called circular, due to the asymptotic distribution of the eigenvalues of \( \frac{1}{\sqrt{n}} A_{ni} \): When \( n \to \infty \), these get uniformly distributed inside the unit circle of the complex plane [37, 38]. (8.10) enables us 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 calculated. In order to calculate \( \phi((a_1 + a_2)^4) \), multiply out \( (a_1 + a_2)^4 \), and use linearity and (8.10) to calculate all \( \phi(a_{i_1} a_{i_2} a_{i_3} a_{i_4}) \) \( (i_j = 1, 2) \). For example, to calculate \( \phi(a_1 a_2 a_1 a_2) \), write it as

\[
\phi((a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I) + \phi(a_1)I)((a_2 - \phi(a_2)I) + \phi(a_2)I)
\]

\[
((a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I) + \phi(a_1)I)((a_2 - \phi(a_2)I) + \phi(a_2)I),
\]

and multiply it out as 16 terms. The term

\[
\phi((a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)
\]

\[
(a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)
\]

is zero by (8.10). The term

\[
\phi((a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I) + \phi(a_1)(a_2 - \phi(a_2)I)
\]

\[
\phi(a_2)(a_1 - \phi(a_1)I)(a_2 - \phi(a_2)I)
\]

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 (8.10) 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 $a_1a_2$ give us two new 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*

$$\mu_1 \boxplus \mu_2$$  \hspace{1cm} (8.12)

for the sum of free random variables, and *multiplicative free convolution*

$$\mu_1 \boxdot \mu_2$$  \hspace{1cm} (8.13)

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 $\mu_1$ and $a_{2n}$ has an eigenvalue distribution which approaches $\mu_2$, 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 8.8.3** Given probability measures $\mu$ and $\mu_2$. When there is a unique probability measure $\mu_1$ such that

$$\mu = \mu_1 \boxplus \mu_2, \mu = \mu_1 \boxdot \mu_2$$ respectively,

we will write

$$\mu_1 = \mu \boxminus \mu_2, \mu_1 = \mu \boxminus \mu_2$$ respectively.

We say that $\mu_1$ is the additive free deconvolution (respectively multiplicative free deconvolution) of $\mu$ with $\mu_2$.

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 $\mu_1$ such that $\mu = \mu_1 \boxplus \mu_2$. For multiplicative free deconvolution, a unique $\mu_1$ 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 Marčhenko Pastur law $\mu_c$ ([39] page 9), also known as the free Poisson distribution in free probability. It is known that $\mu_c$ describes asymptotic eigenvalue distributions of Wishart matrices. Wishart matrices have the form $\frac{1}{N} RR^H$, where $R$ is an $n \times N$ random matrix with independent standard Gaussian entries. $\mu_c$ appears as limits of such when $\frac{n}{N} \rightarrow c$ when $n \rightarrow \infty$. Note that the Marčhenko Pastur law can also hold in the limit for non-gaussian entries.
8.9 Sum of two random matrices

8.9.1 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

$$M_X(s) = \mathbb{E}(e^{sX}), \quad M_{X+Y}(s) = \mathbb{E}(e^{s(X+Y)}).$$

It is then immediate to see that

$$M_X(s) = \frac{M_{X+Y}(s)}{M_Y(s)}.$$

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

$$C_k(X) := \frac{\partial^n}{\partial t^n} \log \left( \mathbb{E} \left( e^{tX} \right) \right) \bigg|_{t=0}.$$

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

$$\forall n \geq 1, m_n(X) = \sum_{p=1}^{n} \sum_{k_1 \geq 1, \ldots, k_p \geq 1 \atop k_1 + \cdots + k_p = n} C_{k_1}(X) \cdots C_{k_p}(X).$$

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.

8.9.2 Additive free convolution

Definition

It is has been proved by Voiculescu ([27]) that for $A_n, B_n$ 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 $A_n, B_n$ converge, as $n$ tends to infinity, to some probability measures $\mu, \nu$, then the eigenvalue distribution of $A_n + B_n$ converges to a probability measure which depends only on $\mu, \nu$, which is called the additive free convolution of $\mu$ and $\nu$, 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 $\rho$ on the real line, which has moments of all order. We shall denote its moments by $m_n(\rho) := \int t^n \, d\rho(t)$, $n \geq 1$. (Note that in the case where $\rho$ is the eigenvalue distribution of a $d \times d$ matrix $A$, these moments can easily be computed by the formula: $m_n(\rho) = \frac{1}{d} \text{Tr}(A^n)$, where $\text{Tr}$ denotes the trace.) We shall associate to $\rho$ another sequence of real numbers, $(K_n(\rho))_{n \geq 1}$, called its free cumulants. The sequences $(m_n(\rho))$ and $(K_n(\rho))$ can be deduced one from each other by the fact that the formal power series $K_\rho(z) := \sum_{n \geq 1} K_n(\rho) z^n$ and $M_\rho(z) := \sum_{n \geq 1} m_n(\rho) z^n$ (8.14) are linked by the relation $K_\rho(z(M_\rho(z) + 1)) = M_\rho(z)$. (8.15)

Equivalently, for all $n \geq 1$, the sequences $(m_0(\rho), \ldots, m_n(\rho))$ and $(K_1(\rho), \ldots, K_n(\rho))$ can be deduced one from each other via the relations

$$m_0(\rho) = 1$$
$$m_n(\rho) = R_n(\rho) + \sum_{k=1}^{n-1} \sum_{l_1, \ldots, l_k \geq 0} m_{l_1}(\rho) \cdots m_{l_k}(\rho)$$

for all $n \geq 1$.

The additive free convolution can be computed easily with the free cumulants via the following characterization (see [40]).

For $\mu, \nu$ compactly supported, $\mu \boxplus \nu$ is the only law $m$ such that for all $n \geq 1$,

$$R_n(m) = R_n(\mu) + R_n(\nu).$$

8.9.3 The additive free deconvolution

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 $\nu$ is (when it exists) the only measure $\mu$ such that $m = \mu \boxplus \nu$. In this case, $\mu$ is denoted by $m \Box \nu$. By theorem 8.9.2, when it exists, $m \Box \nu$ is characterized by the fact that for all $n \geq 1$, $R_n(m \Box \nu) = R_n(m) - R_n(\nu)$.

8.10 Product of two random matrices

8.10.1 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.
8.10.2 The multiplicative free convolution

Definition

It is has been proved by Voiculescu ([27]) that for $A_n, B_n$ 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 $A_n, B_n$ converge, as $n$ tends to infinity, to some probability measures $\mu, \nu$, then the eigenvalue distribution of $A_nB_n$, which is equal to the eigenvalue distribution of $A_1^{1/2}BA_2^{1/2}$ converges to a probability measure which depends only on $\mu, \nu$, which is called the multiplicative free convolution of $\mu$ and $\nu$, 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 $\rho$ 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 $\rho$ another sequence of real numbers, $(s_n(\rho))_{n \geq 0}$, which are the coefficients of what is called its $S$-transform. The sequences $(m_n(\rho))$ and $(s_n(\rho))$ can be deduced one from each other by the fact that the formal power series

$$S_\rho(z) := \sum_{n \geq 1} s_n(\rho) z^{n-1} \text{ and } M_\rho(z) := \sum_{n \geq 1} m_n(\rho) z^n$$

are linked by the relation

$$M_\rho(z) S_\rho(M_\rho(z)) = z(1 + M_\rho(z)). \quad (8.17)$$

Equivalently, for all $n \geq 1$, the sequences $(m_1(\rho), \ldots, m_n(\rho))$ and $(s_1(\rho), \ldots, s_n(\rho))$ can be deduced one from each other via the relations

$$\begin{cases} m_1(\rho)s_1(\rho) = 1, \\ m_n(\rho) = \sum_{l_1 \geq 1} s_{l_1}(\rho) \sum_{l_2 \geq 1} m_{l_1}(\rho) \cdots m_{l_k}(\rho) \prod_{k} l_k = n+1 \end{cases} \quad (8.18)$$

\textbf{rmq 8.10.1} 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 $s_n$ via non crossing partitions and the Kreweras complement, which use more machine time.

\textbf{Example 8.10.2} As an example, it can easily be computed that for the Marchenko-Pastur law $\mu_{\lambda}$ for all $n \geq 1$, $s_n(\mu_{\lambda}) = (-\lambda)^{n-1}$.

The multiplicative free convolution can be computed easily with the free cumulants via the following characterization ([40]).

For $\mu, \nu$ 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 $S_m = S_\rho S_\nu$, i.e. such that for all $n \geq 1$,

$$s_n(m) = \sum_{k+1 \geq n+1} s_k(\mu) s_l(\nu).$$

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 8.10.3.
8.10.3 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 $v$ is (when it exists) the only measure $\mu$ such that $m = \mu \boxtimes v$. In this case, $\mu$ is denoted by $m \boxtimes v$. By theorem 8.10.2, when it exists, $m \boxtimes v$ is characterized by the fact that for all $n \geq 1$,

$$s_n(m \boxtimes v) s_1(v) = s_n(m) - \sum_{k=1}^{n-1} s_k(m \boxtimes v) s_{n-k-1}(v).$$

Hence this operation, very useful to denoise a signal, can be implemented.

8.11 $(M + N)(M + N)^*$

8.11.1 Main result

In this section, we still consider two independent rectangular random matrices $M, N$, 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 $MM^*$ and $NN^*$ (i.e. the uniform distributions on there eigenvalues with multiplicity) both converge to non random probability measures. Here, we shall denote by respectively $\sigma, \tau$ the limit eigenvalue distributions of $MM^*$ and $NN^*$.

Note that in the previously presented results, the case of the limit eigenvalue distribution of $(M + N)(M + N)^*$ 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 normalized 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 $MM^* + NN^*$ or $MM^*NN^*$ (because in these expression, both $M$ and $N$ are always followed by their adjoints), but which is not enough to compute the moments of the eigenvalue distribution of $(M + N)(M + N)^*$. In a recent work ([24]), the first named author generalized Voiculescu’s work to rectangular random matrices, which allowed him to prove that, under the hypothesis made here, the eigenvalue distribution of $(M + N)(M + N)^*$ converges to a probability measure which only depends on $\sigma, \tau$ and $\lambda$, and will be denoted by $\sigma \boxtimes_{\lambda}^\tau$.

rmq 8.11.1 - The symmetric square root of the distribution $\sigma \boxtimes_{\lambda}^\tau$ is be called the rectangular free convolution with ratio $\lambda$ of the symmetric square roots $\sqrt{\sigma}, \sqrt{\tau}$ of $\sigma$ and $\tau$, and denoted by $\sqrt{\sigma \boxtimes_{\lambda}^\tau} \sqrt{\tau}$. The operation $\boxtimes_{\lambda}$ is, rather than $\boxtimes_{\lambda}^\tau$, the one we introduced in [24]. It is essentially equivalent to $\boxtimes_{\lambda}$, as we explain in the footnote below.

As for $\boxtimes$, we shall see two ways to compute $\boxtimes_{\lambda}$: 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.

---

4For any probability measure $\rho$ on $[0, \infty)$, the symmetric square root of $\rho$, 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 $\rho$. Note that $\rho$ 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 symmetrize distributions.
8.11.2 Computing $\boxplus_\lambda$

We fix $\lambda \in [0,1]$. Let us consider a probability measure $\rho$ 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 $\rho$ another sequence of real numbers, $(c_n(\rho))_{n \geq 1}$, depending on $\lambda$, called its rectangular free cumulants with ratio $\lambda$, defined by the fact that the sequences $(m_n(\rho))$ and $(c_n(\rho))$ can be deduced one from each other via the relation

$$C_\rho[z(\lambda M_{\rho^2}(z) + 1)(M_{\rho^2}(z) + 1)] = M_{\rho^2}(z)$$  \hspace{1cm} (8.19)

between the power series

$$C_\rho(z) := \sum_{n \geq 1} c_n(\rho) z^n \text{ and } M_{\rho^2}(z) := \sum_{n \geq 1} m_n(\rho) z^n.$$  \hspace{1cm} (8.20)

Equivalently, for all $n \geq 1$, the sequences $(m_0(\rho), \ldots, m_n(\rho))$ and $(c_1(\rho), \ldots, c_n(\rho))$ can be deduced one from each other via the relations (involving an auxiliary sequence $(m'_0(\rho), \ldots, m'_n(\rho))$)

$$m_0(\rho) = m'_0(\rho) = 1,$$

$$\forall n \geq 1, \quad m'_n(\rho) = \lambda m_n(\rho),$$

$$\forall n \geq 1, \quad m_n(\rho) = c_n(\rho) + \sum_{k=1}^{n-1} \sum_{\substack{l_1, \ldots, l_k \geq 0 \ 1+l_1+\cdots+l_k = n-k}} \prod_{j=1}^{k} c_{l_j}(\rho).$$

**Example 8.11.2** As an example, it is proved in [41] that the law $\mu_\lambda$ has rectangular free cumulants with ratio $\lambda$, 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 characterization ([24]).

**Theorem 8.11.3** For $\sigma, \tau$ compactly supported, $\sigma \boxplus_\lambda \tau$ is the only distribution $m$ such that for all $n \geq 1$, $c_n(m) = c_n(\sigma) + c_n(\tau)$.

### 8.11.3 The rectangular free deconvolution

The moments/cumulants method can also be useful to implement the rectangular free deconvolution. The rectangular free deconvolution with ratio $\lambda$ of a probability measure $m$ on $[0, +\infty)$ by a measure $\tau$ is (when it exists) the only measure $\sigma$ such that $m = \sigma \boxplus_\lambda \tau$. In this case, $\sigma$ is denoted by $m \boxminus_\lambda \tau$. By theorem 8.11.3, 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, very useful to denoise a signal, can be implemented.

---

5Note that again, in [24], these numbers were not called rectangular free cumulants of $\rho$, but of its symmetrized square root.
8.11.4 Discussion

Case where $\lambda = 0$

It is proved in [24] that if $\lambda = 0$, then $\boxplus_0^+ = \boxplus$.

Concretely, it means that if $M, N$ are independent $n \times p$ random matrices which dimensions $n, p$ both tend to infinity such that $n/p \to 0$, then (under the hypothesis that $M$ or $N$ is invariant, in distribution, under multiplication by unitary matrices)

\[
\text{eigenvalue distribution}((M + N)(M + N)^*) \\
\simeq \text{eigenvalue distribution}(MM^* + NN^*).
\]

Case where $\lambda = 1$

It is proved in [24] that if $\lambda = 1$, then for all $\sigma, \tau$ probability measures on $[0, +\infty)$, $\sigma\boxplus_0^+\tau$ is the push forward by the function $t \mapsto t^2$ of the free convolution $\sqrt{\sigma} \boxplus \sqrt{\tau}$ of the symmetrized square roots $\sqrt{\sigma}, \sqrt{\tau}$ of $\sigma$ and $\tau$.

An analytic tool: the rectangular $R$-transform

$\lambda \in [0, 1]$ is still fixed. For $\tau$ probability measure on $[0, +\infty)$, we shall define an analytic function $C_\tau(z)$ in a neighborhood of zero (to be more precise, in a neighborhood of zero in $\mathbb{C} \setminus \mathbb{R}^+$) which, in the case where $\tau$ is compactly supported, has the following series expansion:

\[
C_\tau(z) = \sum_{n \geq 1} c_n(\tau)z^n. \tag{8.21}
\]

It implies, by theorem 8.11.3, that for all compactly supported probability measures $\sigma, \tau$,

\[
C_{\sigma\boxplus_0^+\tau}(z) = C_\sigma(z) + C_\tau(z). \tag{8.22}
\]

Hence the analytic transform $\rho \mapsto C_\rho$ somehow "linearizes" the binary operation $\boxplus_0^+$ on the set of probability measures on $[0, +\infty)$. The analytic function $\tau \mapsto C_\rho$ is called the rectangular $R$-transform with ratio $\lambda$ of $\rho$.

It happens, as we present it below, that for any probability measure $\rho$ on $[0, +\infty)$, $C_\rho$ can be computed in a direct way, without using the definition of (8.21), by the resolution of an equation.

Let us define $M_\rho(z) = \int_{t \in \mathbb{R}^+} \frac{z^t}{\sqrt{t^2 - \lambda}} d\rho(t)$. Then the analytic function $C_\rho$ is defined in a neighborhood of zero (in $\mathbb{C} \setminus \mathbb{R}^+$) to be the solution of

\[
C_\rho\left[z(\lambda M_\rho(z) + 1)(M_\rho(z) + 1)\right] = M_\rho(z), \tag{8.23}
\]

which tends to zero at zero.

To give a more explicit definition of $C_\rho$, let us define

\[
H_\rho(z) = z(\lambda M_\rho(z) + 1)(M_\rho(z) + 1).
\]

\[\text{Again, to make the notations of this paragraph coherent with the ones of the paper [24], where the rectangular machinery was build, one needs to use the duality between measures on }[0, +\infty)\text{ and their symmetrized square roots, which are symmetric measures on the real line.}\]
Then
\[ C_\rho(z) = U \left( \frac{z}{H_\rho^{-1}(z)} - 1 \right) \]

with
\[ U(z) = \begin{cases} \frac{- \lambda - 1 + (\lambda + 1)^2 + 4 \lambda z^2}{\lambda} & \text{if } \lambda \neq 0, \\ z & \text{if } \lambda = 0, \end{cases} \]

where \( z \mapsto z^{1/2} \) is the analytic version of the square root defined on \( \mathbb{C} \setminus \mathbb{R}^- \) such that \( 1^{1/2} = 1 \) and \( H_\rho^{-1} \) is the inverse (in the sense of the composition) of the function \( H_\rho \).

To recover \( \rho \) from \( C_\rho \), 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 what one can easily recover \( \rho \), via its Cauchy transform.

Note that all this method is working for non compactly supported probability measures, and that (8.22) 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 \( \tau, \sigma \). 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 realize practically. However, in the following example, computations are possible.

**Example 8.11.4** 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
\[ \frac{[\kappa^2 - (x - 2)^2]^{1/2}}{\pi \lambda x(4 - x)} \] (8.24)
on its support.

Concretely, it means that if \( A \) is an \( n \times p \) matrix with ones on the diagonal and zeros everywhere else, and \( U, V \) are random \( n \times n, p \times p \) orthogonal matrices with Haar distribution, then as \( n, p \) tend to infinity such that \( n/p \to \lambda \),

\[ \text{eigenvalue distribution}(A + UAV)(A + UAV)^* \]

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 (8.24).

## 8.12 Examples in Wireless Random Networks

### 8.12.1 Topology information

The most simple example is the case where \( f \) is 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 localization 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 $\sigma_k$ 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,

$$y = R^{\frac{1}{2}} s + n,$$

where $R^{\frac{1}{2}} = G\theta$. Here, $G$ is a diagonal matrix with entries $G(f_i)$, $\theta$ is $n \times L$ matrix with entries $e^{-j2\pi f_i \tau_k}$ and $\theta$ is a diagonal matrix with entries $\sigma_k$. $s$ and $n$ 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 $R$ and therefore $\sigma_k$ as suggested in [19].

### 8.12.2 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 $\sigma^2$, 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:

$$y_i = Hs_i + n_i,$$

where $H$ is the $n \times n$ MIMO matrix. The information rate is given by [42]:

$$C = \log \det(I + \frac{1}{\sigma^2} HH^*)$$

$$= \sum_{i=1}^{n} \log(1 + \lambda_i). \quad (8.25)$$

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]:

$$\text{SINR} = \frac{1}{n} \text{trace} \left( HH^* + \sigma^2 \right)^{-1}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\lambda_i + \sigma^2}$$
In both cases, the number of non-zero eigenvalues is also limited to $L$ in general as the medium (matrix $H$) provides only a finite number of degrees of freedom. One can compute these eigenvalues by using the free deconvolution framework on $YY^*$.

### 8.12.3 Power estimation

In TDD heterogenous 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 downlink has a given signature vector of size $n \times 1$ $h_k$ (OFDM, CDMA) with random i.i.d components, the received signal can be written as:

$$y = \sum_{k=1}^{L} h_k \sqrt{P_k} s_k + n$$

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 $y$ and $n$ 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 $YY^*$ as shown in [32].

### 8.13 The Stieljes Transform approach

Let $\mu$ be a probability measure on $\mathbb{R}$. Its Stieljes 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 $\mu$ be compactly supported. Then $G_\mu(z)$ is analytic in a neighborhood of $\infty$. 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 $\mu$ from its Cauchy transform up to a factor. When $\mu$ is absolutely continuous with respect to the Lebesgue measure, its density $f(x)$ is given by:

$$f(x) = + \frac{1}{\pi} \lim_{y \to 0} \text{Im} G_\mu(x + iy)$$

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 moments, $T$ is diagonal with real entries, $A$ is Hermitian and $K/N \to \alpha > 0$ as $N \to \infty$. The behavior 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 $A + WD^H \leq x$). The remarkable result of random matrix theory is the convergence in some sense, of $F^A + WD^H(x)$ 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 8.13.1** Let $A$ be a $N \times N$ hermitian matrix, nonrandom, for which $F^A$ converges weakly as $N \to \infty$ to a distribution function $\mathcal{A}$. Let $F^D$ converges weakly to a nonrandom probability distribution function denoted $\mathcal{D}$. Suppose the entries of $\sqrt{NW}$ 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 $A + WD^H$ converges weakly to a deterministic $F$. Its Stieljes transform $G(z)$ satisfies the equation:

$$G(z) = G_A \left( z - \alpha \int \frac{\tau d\mathcal{T}(\tau)}{1 + \tau G(z)} \right)$$

(8.26)

**8.13.1 A theoretical application example**

We give hereafter the eigenvalue distribution of matrices defined by:

$$W_{N,K}H_{N,K} + \sigma^2 I$$

where $\sqrt{NW}_{N,K}$ is a $N \times K$ matrix with i.i.d entries with zero mean and variance one and $N \to \infty$ such as $\frac{K}{N} \to \alpha$ 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 8.13.1.

Denote $A = \sigma^2 I$ and $D = I_{K,K}$. In this case, $dA(x) = \delta(x - \sigma^2)$ and $dD(x) = \delta(x - 1)$. Applying theorem 8.13.1, the following result is obtained:

$$G(z) = G_{\sigma^2 I} \left( z - \alpha \int \frac{\delta(\tau - 1) d\tau}{1 + \tau G(z)} \right)$$

(8.27)

$$= G_{\sigma^2 I} \left( z - \frac{\alpha}{1 + G(z)} \right)$$

(8.28)

$$= \int \frac{\delta(\sigma^2 - \lambda) d\lambda}{\lambda - z + \frac{\alpha}{1 + G(z)}}$$

(8.29)

$$= \frac{1}{\sigma^2 - z + \frac{\alpha}{1 + G(z)}}$$

(8.30)

$G_{\sigma^2 I}(z)$ is the Cauchy transform of the eigenvalue distribution of matrix $\sigma^2 I$. The solution of the second order equation 8.30 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 have plotted the theoretical and practical eigenvalue distribution for various size matrices. Only one realization of matrix $W_{N,K}W_{N,K}^H + \sigma^2I$ has been studied. As the dimensions increase, the practical results fit extremely well the theoretical results as shown in fig.A.1 and fig.A.2.

Figure 8.4: matrix size: 256*64, no noise, alpha=0.25.

Figure 8.5: matrix size: 4096*1024, 10dB, alpha=0.25.
## 8.13.2 A wireless communication example

Most of the information theoretic literature focuses on vector memoryless channels of the form:

\[ Y = Hs + n. \]  

(8.31)

Eq. (8.31) 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

\[
\frac{1}{N} C = \frac{1}{N} \log \det \left( I + \frac{1}{\sigma^2} HH^H \right) 
= \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + \frac{1}{\sigma^2} \lambda_i \left( HH^H \right) \right) 
= \int \log \left( 1 + \frac{1}{\sigma^2} \lambda \right) \frac{1}{N} \sum_{i=1}^{N} \delta \left( \lambda - \lambda_i \left( HH^H \right) \right) d\lambda 
= \int \log \left( 1 + \frac{1}{\sigma^2} \lambda \right) F_{HH^H}(\lambda) d\lambda.
\]

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 multiuser 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:

\[
\frac{1}{N} C = \int \log \left( 1 + \frac{1}{\sigma^2} \lambda \right) F_{HH^H}(\lambda) d\lambda,
\]

and differentiating according to \( \sigma^2 \), we obtain

\[
\frac{1}{N} \frac{\partial C}{\partial \sigma^2} = \int \frac{-1}{\sigma^4} \frac{\lambda}{1 + \frac{1}{\sigma^2} \lambda} F_{HH^H}(\lambda) d\lambda 
= -\frac{1}{\sigma^4} \int \frac{\lambda + 1 - 1}{\sigma^2 \lambda + 1} F_{HH^H}(\lambda) d\lambda 
= -\frac{1}{\sigma^4} + \int \frac{1}{\lambda + \sigma^2} F_{HH^H}(\lambda) d\lambda 
= -\frac{1}{\sigma^4} + m_{HH^H}(-\sigma^2).
\]

Hence, finding the Stieltjes transform is often enough.
Bibliography


Chapter 9

Tools from Physics and Road-traffic Engineering for Dense Ad-hoc Networks

E. Altman, P. Bernhard, M. Debbah, A. Silva

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 minimize its individual cost. We seek for 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 optimization problem, (2) a method based on Green 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 optimization problem.

9.1 Introduction

In the design and analysis of wireless networks, researchers frequently stumble on the scalability problem that can be summarized 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 optimization 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] 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.

The physical paradigms allow the authors to minimize 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 minimization) 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. 9.1 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 \((0,0)\) and a destination node placed at the location \((0,200)\), through an area where relay nodes are placed according to a spatial Poisson process of density \(\lambda(x) = a \cdot [10^{-4}x^2 + 0.05]\) 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 \(\delta\) 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

\footnote{We note that this approach is restricted to costs that do not depend on the congestion}
precise node placement, but only on the node density function \( \lambda(x) \) 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 characterization 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 optimization models in routing or to node assignment. We then formulate the global optimization problem and the individual optimization one with a focus on the directional antennas scenario. We provide several approaches to obtain both qualitative characterization as well as quantitative solutions to the problems.

### 9.2 Determining routing costs in dense ad-hoc networks

In optimizing a routing protocol in ad-hoc networks, or in optimizing 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.

#### 9.2.1 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 \( \lambda \). 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 \( \lambda \). A typical cost (see e.g. [25]) at a neighborhood of \( x \) is the density of nodes required there to carry a given flow. Assuming that a flow \( T(x) \) is assigned through a neighborhood of \( x \), the cost is taken to be

\[
c(x, T(x)) = f^{-1}(|T(x)|)
\]

We denote with bold font the vectors.
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 \( \Omega(\frac{\sqrt{\lambda}}{\log \lambda}) \). Using percolation theory, the authors of [9] have shown that in the randomly located set the same \( \Omega(\sqrt{\lambda}) \) can be achieved.

- Baccelli, Blaszczyszyn 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

\[
c(T(x)) = k|T(x)|^2
\]

This follows from (9.1) as \( f(x) \) behaves like \( \sqrt{x} \) so its inverse is quadratic.

### 9.2.2 Congestion independent routing

A metric often used in the Internet for determining routing is the number of hops, which routing protocol try to minimize. The number of hops is proportional to the expected delay along the path in the context of ad-hoc networks, in case that the queuing 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 \( c(x) \) is constant so that the cost of a path is its length in meters. The routing then follows a shortest path selection.

- Let us assume that the range \( R(x) \) depends on local radio conditions at a point \( x \) (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 minimization problem. We further assume, as in Gupta and Kumar, that the range is scaled to go to 0 as the total density \( \lambda \) of nodes grows to infinity. More precisely, let us consider a scaling of the range such that the following limit exists:

\[
r(x) := \lim_{\lambda \to \infty} \frac{R(x)}{\lambda}
\]

Then in the dense limit, the fraction of nodes that participate in forwarding packets along a path is \( 1/r(x) \) 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 equalize the hop distance.

9.2.3 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 optimized so as to minimize 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 9.2.1) 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
\[ f(\lambda) = \Omega \left( \frac{\lambda}{\log \lambda} \right)^{q-1} \] where \( q \) is the path-loss, see [21]. The cost is then obtained using (9.1).

9.3 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 \( \rho(x) \), measured in bps/m², such that at locations \( x \) where \( \rho(x) > 0 \) there is a distributed data source such that the rate with which information is created in an infinitesimal area of size \( d\Omega \) centered at \( x \) is \( \rho(x)d\Omega \). Similarly, at locations \( x \) where \( \rho(x) < 0 \) there is a distributed data sink such that the rate with which information is absorbed by an infinitesimal area of size \( d\Omega \), centered at point \( x \), is equal to \(-\rho(x)d\Omega\).

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.
\[
\int_{X_1 \times X_2} \rho(x)dS = 0.
\]

Next we present the flow conservation condition (see e.g. [25, 6] for more details). For information to be conserved over a domain \( \Omega_0 \) 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.
\[
\int_{\Omega_0} \rho(x)dx = \oint_{\partial\Omega_0} [T(x) \cdot n(x)]d\ell
\]

The integral on the left is the surface integral of \( \rho(x) \) over \( \Omega_0 \). The integral on the right is the path integral of the inner product \( T \cdot n \) over the curve \( \partial\Omega_0 \). The vector \( n(x) \) is the unit normal vector to \( \partial\Omega_0 \) at the boundary point \( x \in \partial\Omega_0 \) and pointing outwards. The function \( T(x) \cdot n(x) \) measured in bps/m² is equal at the rate with which information is leaving the domain \( \Omega_0 \) per unit length of boundary at the boundary point \( x \).

This holding for any (smooth) domain \( \Omega_0 \), it follows that necessarily
\[
\nabla \cdot T(x) := \frac{\partial T_1(x)}{\partial x_1} + \frac{\partial T_2(x)}{\partial x_2} = \rho(x),
\]

(9.3)
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 $a$ to a point $b$. 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 $\nu$ traffic classes. Extending the relation (9.3) to the multi-class case, we have traffic conservation at each point in space to each traffic class as expressed in the following:

$$\nabla \cdot T_j(x) = \rho_j(x), \quad \forall x \in \Omega. \quad (9.4)$$

The function $T_j$ is the flow distribution of class $j$ and $\rho_j$ corresponds to the distribution of the external sources and/or sinks.

Let $T(x)$ be the total flow vector at point $x \in \Omega$. A generic multi-class optimization problem would then be: minimize $Z$ over the flow distributions $\{T_j\}$

$$Z = \int_{\Omega} g(x, T(x)) dx_1 dx_2 \text{ subject to } \nabla \cdot T_j(x) = \rho_j(x), \quad j = 1, \ldots, \nu \quad \forall x \in \Omega. \quad (9.5)$$

### 9.4 Directional Antennas and Global Optimization

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 $\nu$ classes of flows $T_i^j \geq 0$, $T_i^j \geq 0$, $j = 1, \ldots, \nu$. 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 $x$ to be different than the cost for a vertical transmission (North-South). It is assumed that a packet located at the point $x$ and traveling in the direction of the axis $x_i$ incurs a transportation cost $g_i$ and such transportation cost depends upon the position $x$ and the traffic flow $T(x)$. We thus allow for a vector valued cost $g := g(x, T(x))$. 

The local cost corresponding to the global optimization problem is given by \( g(x, T(x)) = g(x, T(x)) \cdot T(x) \) 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 \( g(x, T(x)) \) is assumed to be non-negative, convex increasing in each of the components of \( T \) (\( T_1 \) and \( T_2 \) in our 2-dimensional case).

The boundary conditions will be determined by the options that travelers 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 minimize \( Z \) as defined in (9.5). 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 \( T_j \) in \( L^2(\Omega) \), so that \( \rho \) may be in \( H^{-1}(\Omega) \), allowing for some localized mass of traffic source or sink.

**Kuhn-Tucker conditions.** Define the Lagrangian as

\[
L^\xi(x, T) := \int_\Omega \ell^\xi(x, T) \, dx \quad \text{where } \ell^\xi(x, T) = g(x, T(x)) - \sum_{j=1}^y \zeta^j(x) \left[ \nabla \cdot T^j(x) - \rho^j(x) \right]
\]

where the \( \zeta^j(x) \in H^1(\Omega) \) are Lagrange multipliers. The criterion is convex, and the constraint (9.4) affine. Therefore the Kuhn-Tucker theorem holds, stating that the Lagrangian is minimum at the optimum. A variation \( \delta T(\cdot) \) will be admissible if \( T(x) + \delta T(x) \geq 0 \) for all \( x \), hence in particular, \( \forall x : T_j^i(x) = 0, \delta T_j^i(x) \geq 0 \).

\[
\forall \delta T \text{ admissible, } DL^\xi \cdot \delta T \geq 0,
\]

therefore here

\[
\int_\Omega \sum_j \langle \nabla T_j g(x, T(x)), \delta T^j(x) \rangle \, dx - \int_\Omega \sum_j \zeta^j(x) \nabla \cdot \delta T^j(x) \, dx \geq 0.
\]

Integrating by parts with Green’s formula, this is equivalent to

\[
\int_\Omega \sum_j \left[ \langle \nabla T_j g, \delta T^j \rangle + \langle \nabla_x \zeta^j, \delta T^j \rangle \right] \, dx - \int_{\partial \Omega} \sum_j \zeta^j \langle \delta T^j, n \rangle \, d\ell \geq 0.
\]

We may choose all the \( \delta T^k = 0 \) except \( \delta T^j \), 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 \( i = 1, 2 \):
\[
\frac{\partial g(x, T)}{\partial T_i} + \frac{\partial \zeta_j / \partial x_i}{\partial x_i} = 0 \quad \text{if } T_i(x) > 0 \quad (9.6a)
\]
\[
\frac{\partial g(x, T)}{\partial T_i} + \frac{\partial \zeta_j / \partial x_i}{\partial x_i} \geq 0 \quad \text{if } T_i(x) = 0. \quad (9.6b)
\]
Placing this back in Euler’s inequality, and using a \( \delta T \) non zero on the boundary, it follows that necessarily \( \zeta_j(x) = 0 \) at any \( x \) of the boundary \( \partial \Omega \) where \( T(x) > 0 \). This provides the boundary condition to recover \( \zeta_j \) from the condition (9.4).

Remark: The Kuhn-Tucker type characterization (9.6a)-(9.6b) 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
\[
g(x, T(x)) = \sum_{i=1,2} g_i(x, T(x)) T_i(x).
\]

1. Monomial cost per packet:
\[
g_i(x, T(x)) = k_i(x) \left( T_i(x) \right)^\beta \quad (9.7)
\]
for some \( \beta > 1 \). Then (9.6a)-(9.6b) simplify to
\[
(\beta + 1) k_i(x) \left( T_i(x) \right)^\beta + \frac{\partial \zeta_j (x)}{\partial x_i} = 0 \quad \text{if } T_i(x) > 0 \quad (9.8a)
\]
\[
(\beta + 1) k_i(x) \left( T_i(x) \right)^\beta + \frac{\partial \zeta_j (x)}{\partial x_i} \geq 0 \quad \text{if } T_i(x) = 0. \quad (9.8b)
\]
In that case, recovery of \( \zeta \) to complete the process is difficult, at best. Things are simpler in the next case.

2. Affine cost per packet:
\[
g_i(x, T(x)) = \frac{1}{2} k_i(x) T_i(x) + h_i(x). \quad (9.9)
\]
Then (9.6a)-(9.6b) simplify to
\[
k_i(x) T_i(x) + h_i(x) + \frac{\partial \zeta_j (x)}{\partial x_i} = 0 \quad \text{if } T_i(x) > 0
\]
\[
k_i(x) T_i(x) + h_i(x) + \frac{\partial \zeta_j (x)}{\partial x_i} \geq 0 \quad \text{if } T_i(x) = 0.
\]
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 \( T(x) > 0 \) for all \( x \). Then
\[
T_i(x) = - \left( a_i(x) \frac{\partial \zeta_j (x)}{\partial x_i} + b_i(x) \right).
\]
As a consequence, from (9.4) 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 \( H^{-1}(\Omega) \))
\[
\sum_i \frac{\partial}{\partial x_i} \left( a_i(x) \frac{\partial \zeta_j (x)}{\partial x_i} + \nabla \cdot b(x) + p(x) = 0. \right.
\]
\[
^3 \text{This is a complementary slackness condition on the boundary.}
\]
This is a well behaved Dirichlet problem, known to have a unique solution in $H^1_0(\Omega)$, furthermore easy to compute numerically.

9.5 User optimization and congestion independent costs

We expand on the shortest path approach for optimization that has already appeared using geometrical optics tools [17]. We present general optimization frameworks for handling shortest path problems and more generally, minimum cost paths.

We consider the model of Section 9.4. We assume that the local cost depends on the direction of the flow but not on its size. The cost is $c_1(x)$ for a flow that is locally horizontal and is $c_2(x)$ for a flow that is locally vertical. We assume in this section that $c_1$ and $c_2$ do not depend on $T$. The cost incurred by a packet transmitted along a path $p$ is given by the line integral

$$c_p = \int_p c \cdot dx.$$  \hspace{1cm} (9.11)

Let $V^j(x)$ be the minimum cost to go from a point $x$ to a set $B^j$, $j = 1, \ldots, v$. Then

$$V^j(x) = \min \left( c_1(x)dx_1 + V^j(x_1 + dx_1, x_2), c_2(x)dx_2 + V^j(x_1, x_2 + dx_2) \right)$$ \hspace{1cm} (9.12)

This can be written as

$$0 = \min \left( c_1(x) + \frac{\partial V^j(x)}{\partial x_1}, c_2(x) + \frac{\partial V^j(x)}{\partial x_2} \right), \quad \forall x \in B^j, \quad V^j(x) = 0,$$ \hspace{1cm} (9.13)

If $V^j$ is differentiable then, under suitable conditions, it is the unique solution of (9.13). In the case that $V^j$ is not everywhere differentiable then, under suitable conditions, it is the unique viscosity solution of (9.13) (see [3, 8]).

There are many numerical approaches for solving the HJB equation. One can discretize the HJB equation and obtain a discrete dynamic programming for which efficient solution methods exist. If one repeats this for various discretization 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].

9.6 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 9.4).

**Definition 9.6.1** [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 $\Omega$, defined by the simple closed curve $\partial R^+ = \Gamma_1^+ \cup \Gamma_2^+ \cup \Gamma_3^- \cup \Gamma_4^-$ (see Fig. 9.2).

We obtain below optimal paths defined as paths that achieve the minimum cost in (9.11). 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

$$U(x) = \frac{\partial c_2}{\partial x_1}(x) - \frac{\partial c_1}{\partial x_2}(x) \quad \forall x \in \Omega.$$

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 $c \in C^1(\Omega)$ then $U$ is a continuous function on $\Omega$. This motivates us to study cases in which $U$ has the same sign everywhere (see Fig. 9.3), 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. 9.4).

\[
\begin{align*}
\Gamma_1^+ &= \{0 \leq x_1 \leq a, \ x_2 = 0\} \\
\Gamma_2^+ &= \{x_1 = a, \ 0 \leq x_2 \leq b\} \\
\Gamma_3^- &= \{0 \leq x_1 \leq a, \ x_2 = b\} \\
\Gamma_4^- &= \{x_1 = 0, \ 0 \leq x_2 \leq b\}.
\end{align*}
\]

Figure 9.2: The boundaries of the region $R$.

\[
\begin{align*}
\Gamma_1^- &= \{0 \leq x_1 \leq a, \ x_2 = 0\} \\
\Gamma_2^- &= \{x_1 = a, \ 0 \leq x_2 \leq b\} \\
\Gamma_3^- &= \{0 \leq x_1 \leq a, \ x_2 = b\} \\
\Gamma_4^- &= \{x_1 = 0, \ 0 \leq x_2 \leq b\}.
\end{align*}
\]

Figure 9.4: The case of two regions separated by a curve. Case 1.

We shall assume throughout that the function $c \in C^1(\Omega)$, and that, if non empty, the set $M = \{x \mid U(x) = 0\}$ is a smooth line. (This is true, e.g., if $c \in C^2$ and $\nabla U \neq 0$ on $M$.)

**9.6.1 The function $U$ has the same sign over the whole region $R$**

**Theorem 9.6.1** (Point to point optimal path) Suppose that a point $x^O = (x_1^O, x_2^O)$ in $\overset{\bullet}{R}$ (the interior of $R$), wants to send a packet to a point $x^D = (x_1^D, x_2^D)$ in $\overset{\bullet}{R}$.
i. If \( U > 0 \) in the region \( R_{OD} = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^O, x_2^O \leq x_2 \leq x_2^O \} \), except perhaps from a set of Lebesgue measure zero, then there is an optimal path given by (see Fig. 9.5):

\[
\gamma^{opt} = \gamma_H \cup \gamma_V \text{ where }
\]

\[
\gamma_H = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^D, x_2 = x_2^O \}
\]

\[
\gamma_V = \{(x_1, x_2) \text{ such that } x_1 = x_1^D, x_2^O \leq x_2 \leq x_2^D \}.
\]

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. 9.6):

\[
\gamma^{opt} = \gamma_V \cup \gamma_H \text{ where }
\]

\[
\gamma_V = \{(x_1, x_2) \text{ such that } x_1 = x_1^O, x_2^O \leq x_2 \leq x_2^D \}
\]

\[
\gamma_H = \{(x_1, x_2) \text{ such that } x_1^O \leq x_1 \leq x_1^D, x_2 = x_2^D \}.
\]

iii. In both cases, \( \gamma^{opt} \) is unique up to a zero Lebesgue measure. (i.e. the Lebesgue measure of the area between \( \gamma^{opt} \) and any other optimal path is zero).

**Proof.**- Consider an arbitrary path\(^4\) \( \gamma_C \) joining \( x^O \) to \( x^D \), and assume that the Lebesgue measure of the area between \( \gamma^{opt} \) and \( \gamma_C \) is nonzero. We call such path, the comparison path (see Fig. 9.5 for the case \( U > 0 \) and Fig. 9.6 for \( U < 0 \)).

(i) Showing that the cost over path \( \gamma^{opt} \) is optimal is equivalent to showing that the integral of the cost over the closed path \( \xi^- \) is negative, where \( \xi^- \) is given by following \( \gamma^{opt} \) from the source \( x^O \) to the destination

\(^4\)Respecting that each subpath 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.
\(x^D\) and then returning from \(x^D\) to \(x^O\) by moving along the path \(\gamma_C\) in the reverse direction. This closed path is written as \(\xi^- = \gamma^+_C \cup \gamma^-_C \cup \xi^-\) and \(\Omega_1\) denotes the bounded area described by \(\xi^-\). Using Green Theorem (see Appendix) we obtain

\[
\oint_{\xi^-} c \cdot dx = -\int_{\Omega_1} U(x) dS
\]

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 9.6.2 (Point to boundary optimal path)

Consider the problem of finding an optimal path from a point \(\bar{x} \in \mathring{R}\) to the boundary \(\Gamma_1 \cup \Gamma_2\).

i. Assume that \(U(x) < 0\) for all \(x \in \mathring{R}\) except perhaps for a set of Lebesgue measure zero. Assume that the cost on \(\Gamma_1\) is non-negative and that the cost on \(\Gamma_2\) is non-positive. Then the optimal path is the straight vertical line.

ii. Assume that \(U(x) > 0\) for all \(x \in \mathring{R}\) except perhaps for a set of Lebesgue measure zero. Assume that the cost on \(\Gamma_1\) is non-positive and that the cost on \(\Gamma_2\) is non-negative. Then the optimal path is the straight horizontal line.

Proof.-

(i) Denote by \(\gamma_V\) the straight vertical path joining \(\bar{x}\) to \(\Gamma_1\). Consider another arbitrary valid path \(\gamma_C\) joining \(\bar{x}\) to any point \(x^*\) on \(\Gamma_1 \cup \Gamma_2\), and assume that the Lebesgue measure of the area between \(\gamma_{opt}\) and \(\gamma_C\) is nonzero. We call such path, the comparison path.

Assume first that \(x^*\) is on \(\Gamma_2\). Denote \(x^D := \Gamma_1 \cap \Gamma_2\). Then by Theorem 9.6.1 (ii), the cost to go from \(\bar{x}\) to \(x^D\) is smaller when using \(\gamma_V\) and then continuing eastwords (along \(\Gamma_1^+\)) than when using \(\gamma_C\) and then southwords (along \(\Gamma_2^-\)). Due to our assumptions on the costs over the boundaries, this implies that the cost along \(\gamma_V\) is smaller than along \(\gamma_C\).

Next consider the case where \(x^*\) is on \(\Gamma_1\). Denote by \(\eta\) the section of the boundary \(\Gamma_1\) that joins \(\gamma_V \cap \Gamma_1\) with \(x^*\) (see Figure 9.7). Then again, by Theorem 9.6.1 (ii), the cost to go from \(\bar{x}\) to \(x^*\) is smaller when using \(\gamma_V\) and then continuing eastwords (along \(\Gamma_1^+\)) than when using \(\gamma_C\). Due to our assumptions that the cost on \(\Gamma_1\) is non-negative, this implies that the cost along \(\gamma_V\) is smaller than along \(\gamma_C\).

(ii) is obtained similarly.

9.6.2 The function \(U\) changes sign within the region \(R\)

Consider the region on the space \(M := \{x \in \Omega \text{ such that } U(x) = 0\}\). 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. \(U(x)\) is negative in the upper area and positive in the lower area (see Fig. 9.4).
2. \(U(x)\) is positive in the upper area and negative in the lower area (see Fig. 9.8).
Figure 9.7: Theorem 9.6.2 (i)

Two other cases where the sign of $U$ is the same over $\Omega$ 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 $U(x)$ is negative in the upper area and positive in the lower area.

We shall show that this case, $M$ is an attractor.

**Proposition 9.6.1** Assume that the source $x$ and destination $y$ are both on $M$. Then the path $p_M$ that follows $M$ is optimal.

*Proof.* Consider an alternative path $\gamma_C$ that coincides with $M$ only in the source and destination points. First assume $\gamma_C$ is entirely in the upper (i.e. northern) part and call $\Omega_1$ the surrounded area. Define $\xi^+$ to be the closed path that follows $p_M$ from $x$ to $y$ and then returns along $\gamma_C$.

The integral $\int_{\Omega_1} U(x) dS$ is negative by assumption. By Green Theorem it equals $\oint_{\xi^+} c \cdot dx$. This implies that the cost along $p_M$ is strictly smaller than along $\gamma_C$.

A similar argument holds for the case that $\gamma_C$ is below $p_M$.

A path between $x$ and $y$ may have several intersections with $M$. Between each pair of consecutive intersections of $M$, the subpath 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 9.6.2** Let a point $x^O$ send packets to a point $x^D$.

(i) Assume both points in the upper region. Denote by $\gamma_1$ the two segments path given in Theorem 9.6.1 (ii). Then the curve $\hat{\gamma}$ obtained as the maximum between $M$ and $\gamma_1$ is optimal.\(^5\)

(ii) Let both points be in the lower region. Denote by $\gamma_2$ the two segments path given in Theorem 9.6.1 (i). Then the curve $\hat{\gamma}$ obtained as the minimum between $M$ and $\gamma_2$ is optimal.

\(^5\)By the maximum we mean the following. If $\gamma_1$ does not intersect $M$ then $\hat{\gamma} = \gamma_1$. If it intersects $M$ then $\hat{\gamma}$ agrees with $\gamma_1$ over the path segments where $\gamma_1$ is in the upper region and otherwise agrees with $M$. The minimum is defined similarly
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 $\gamma_C$ that is not restricted to the upper region. Then $M \cap \gamma_C$ contains two distinct points such that $\gamma_C$ is strictly lower than $M$ between these points. Applying Proposition 9.6.1 we then see that the cost of $\gamma_C$ can be strictly improved by following $M$ between these points instead of following $\gamma_C$ there. This concludes (i). (ii) is proved similarly.

Proposition 9.6.3 Let a point $\bar{x}^O$ send packets to a point $x^D$.

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 $\bar{x}^O$ 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 $x^D$.

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 $\bar{x}^O$ 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 $x^D$.

Proof. The proofs of (i) and of (ii) are the same. Consider an alternative route $\gamma_C$. Let $\tilde{x}$ 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 $\tilde{x}$ and second, the optimal path between $\tilde{x}$ and the destination.

Case 2: The function $U$ is positive in the upper area and negative in the lower area.

![Diagram of two regions separated by the curve $M$. Case 2.](image)
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.

**Proposition 9.6.4** Assume that both source and destination are in the same region. Then the paths that are optimal in Theorem 9.6.1 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 9.6.1 which we can then apply.

**Discussion.** Note that the (sub)optimal policies obtained in Proposition 9.6.4 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.

### 9.7 User optimization 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 decentralized 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 decentralized 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 optimization 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 equalizing source-destination delays of packets that belong to the same class, which allows one to minimize 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 9.4. Let $V^j(x)$ be the minimum cost to go from a point $x$ to $B^j$ at equilibrium. Equation (9.12) still holds but this time with $c_i$ that depends on $T_j^i$, $i = 1, 2$, and on the total flows $T_i^j$, $i = 1, 2$. Thus (9.13) becomes, $\forall j \in \{1, \ldots, \nu\}$,

$$0 = \min_{i=1,2} \left( c_i(x, T_i) + \frac{\partial V^j(x)}{\partial x_i} \right), \quad \forall x \in B^j, V^j(x) = 0. \quad (9.14)$$

We note that if $T_i^j(x) > 0$ then by the definition of the equilibrium, $i$ attains the minimum at (9.14). Hence (9.14) implies the following relations for each traffic class $j$, and for $i = 1, 2$:

$$c_i(x, T_i) + \frac{\partial V^j}{\partial x_i} = 0 \quad \text{if} \quad T_i^j > 0, \quad (9.15a)$$

$$c_i(x, T_i) + \frac{\partial V^j}{\partial x_i} \geq 0 \quad \text{if} \quad T_i^j = 0. \quad (9.15b)$$

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 minimization one. We shall restrict here to the single class case. To that end, we note that equations (9.15a)-(9.15b) have exactly the same form as the Kuhn-Tucker conditions (9.6a)-(9.6b), except that $c_i(x, T_i)$ in the former are replaced by $\partial g(x, T_i)/\partial T_i(x)$ in the latter. We therefore introduce a potential function $\psi$ defined by

$$\psi(x, T) = \sum_{i=1,2} \int_0^{T_i} c_i(x, s) ds$$

so that for both $i = 1, 2$:

$$c_i(x, T_i) = \frac{\partial \psi(x, T)}{\partial T_i}.$$ 

Then the user equilibrium flow is the one obtained from the global optimization problem where we use $\psi(x, T)$ as local cost. Hence, the Wardrop equilibrium is obtained as the solution of

$$\min_{T_i} \int_{\Omega} \psi(x, T) dx \quad \text{subject to} \quad \nabla \cdot T(x) = \rho(x), \quad \forall x \in \Omega.$$ 

In the special case where costs are given as a power of the flow as defined in eq. (9.7), we observe that equations (9.15a)-(9.15b) coincide with equations (9.8a)-(9.8b) (up-to a multiplicative constant of the cost). We conclude that for such costs, the user equilibrium and the global optimization solution coincide.

### 9.8 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 9.4. 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 optimization 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 minimize that users cost among all its possible choices.

Consider the rectangular area $R$ on the bounded domain $\Omega$ defined by the simple closed curve $\partial R = \Gamma_1^+ \cup \Gamma_2^+ \cup \Gamma_3^- \cup \Gamma_4^-$ where

$$\Gamma_1 = \{0 \leq x_1 \leq a, \quad x_2 = 0\}, \quad \Gamma_2 = \{x_1 = a, \quad 0 \leq x_2 \leq b\},$$

$$\Gamma_3 = \{0 \leq x_1 \leq a, \quad x_2 = b\}, \quad \Gamma_4 = \{x_1 = 0, \quad 0 \leq x_2 \leq b\}.$$

Assume throughout that $\rho = 0$ for all $x \in \partial \Omega$, and that the costs of the routes are linear, i.e.

$$c_1 = k_1 T_1 + h_1 \quad \text{and} \quad c_2 = k_2 T_2 + h_2,$$

with $k_1 > 0, k_2 > 0, h_1$, and $h_2$ constant over $\Omega$.

We are precisely in the framework of section 9.7 and 9.4 with affine costs per packet. As a matter of fact, the potential function associated with these costs is

$$\psi(T) = \sum_{i=1}^2 \int_0^{T_i} (k_i s + h_i) ds = \sum_{i=1}^2 \frac{1}{2} k_i T_i + h_i T_i.$$

Now, we want to handle a condensation of sources or sinks along the boundary. While this is feasible with the framework of section 9.4, it is rather technical. We rather use a more direct path below.

Notice that we have in $\partial \Omega$, we have

$$\frac{\partial T_1}{\partial x_2} + \frac{\partial T_2}{\partial x_1} = 0.$$

Take any closed path $\gamma$ surrounding a region $\omega$. Then by Green formula,

$$\oint_{\gamma} T_1 d\xi_2 - T_2 d\xi_1 = \int_\omega \frac{\partial T_1}{\partial x_2} + \frac{\partial T_2}{\partial x_1} = 0.$$

Therefore we can define

$$\phi(x) := \int_\omega T_1 d\xi_2 - T_2 d\xi_1$$

the integral will not depend on the path between $x^o$ and $x$ and $\phi$ is thus well defined, and we have

$$\frac{\partial \phi(x)}{\partial x_2} = T_1(x) \quad \frac{\partial \phi(x)}{\partial x_1} = -T_2(x).$$

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 $\Omega$ [6]. It turns out that all paths to the destination are used. Thus, from Wardrop’s principle, the cost $\int c \, dx$ is equalized between any two paths. And therefore,

$$\frac{\partial c_1}{\partial x_2} = \frac{\partial c_2}{\partial x_1},$$

Using the equations in (9.16) then

$$k_1 \frac{\partial T_1}{\partial x_2} = k_2 \frac{\partial T_2}{\partial x_1},$$

and from equations in (9.17) 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 analyzing 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 in 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, \( \phi \) may be the sum of an arbitrary number of such multiplicative decompositions with different \( s \). We therefore arrive at general formula such as
\[
\phi(x_1, x_2) = \int [A(s) \cos(isK_1 x_1) + B(s) \sin(isK_1 x_1)][C(s) \cos(sK_2 x_2) + D(s) \sin(sK_2 x_2)] ds.
\]

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 formulas we obtain for the \( T_i \)’s.)

### 9.9 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 optimization frameworks.

### Acknowledgement

We wish to thank Dr. Stavros Toumpis for helpful discussions. The work has been kindly supported by the Bionets European Contract.
Bibliography


209


9.10 Appendix: Mathematical Tools

Theorem 9.10.1 (Green’s Theorem) Let $\Omega \subseteq X$ be a region of the space, and let $\Gamma$ 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 $\Omega$ ) Then

$$\oint_{\Gamma} Pdx + Qdy = \int_{\Omega} \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy.$$  (9.18)
Part III

Socio-Economic Paradigms
Chapter 10

Historical-Interpretive Considerations about Money as a Unit of Value and Scale-Dependent Phenomenon

S. Elaluf-Calderwood

10.1 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. These models have been actively used over time but since economic interconnections tend to be of a socio-geographical nature (Schraven, 2001), the adoption of such models has been highly ideologically motivated, such as in the case of the Local Exchange and Trading Systems (LETS) (Cadwell, 2000). Membership of these types of economic initiatives has been exclusive of the average population and not economically motivated, and is sustained by considerable social effort in the medium and long term (Schraven, 2001). A similar situation occurs in the case of complementary currencies, which can coexist with national currencies.

The rise of mobile technology and the multiple applications and services that can be provided directly to users offers the opportunity to liberate the economics of sharing and community currencies from the ideological trap, 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 for how to understand, apply and negotiate 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? (The Economist, 2005).

This paper is structured in five sections. The first section discusses a global abstraction 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. 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 ends this paper.

10.2 Historical Background

Money has a social interpretive value and this is clear if seeking a definition of money using neo-classical economics. In the Economics field there are several definitions of money; some of these definitions overlap others and others have distinct semantic boundaries. In this document 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 over other categorizations. 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 (Wikipedia, 2006). In everyday use money refers more specifically to currency, and the circulation of currencies with a legal tender status, which works as a standard of deferred payment. Historically 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 (Menger, 1982).

Modern barter models deserve a more in-depth analysis in light of the fact that it has been 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 to. 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.

Figure 1 is an illustration of a medieval Sunday Village Fair. 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 European Medieval village is a rich source of metaphors for community currencies and economics of sharing. 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.

![The Medieval Village: Barter economy and multiple community currencies in action](image)

In the Medieval Village there is limited centralization when issuing money or currency. There is trust between villagers or group of villagers at trading time 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 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 money object that has negligible inherent value. The widespread acceptance of Fiat Money is most frequent when the central authority mandates the acceptance of money under penalty of law and demands this money in payment of taxes or tribute.

According to this centralized view, money needs to be provided with some essential value characteristics. The
characteristics are (Menger, 1982):

- 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—indeed be predictably useful when it is so retrieved.

To acquire these characteristics, Fiat Money should have a number of other features (Cesarano, 2006). See Table 1.

This definition of money is focused on expressing money as a numerical value and scale-dependent phenomenon—generalization of its use—ignoring potential aspects of money as a social exchange trading tool. It is difficult to envision 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 to 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.

Table 1. Characteristics of money

<table>
<thead>
<tr>
<th>Money as a medium of exchange</th>
<th>Money as a unit of account</th>
<th>Money as a store of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. It should have liquidity, and be easily tradable, with a low spread between the prices to buy and sell, in other words, a low transaction cost.</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
<td>2. It should be fungible; that is, one unit or piece must be exactly equivalent to another.</td>
<td>2. It should have a stable value.</td>
</tr>
<tr>
<td>3. It should be physically durable and non-health hazardous.</td>
<td>3. It must be a specific weight, or measure, or size to be verifiably countable</td>
<td>3. It should be difficult to counterfeit, and the genuine must be easily recognizable.</td>
</tr>
</tbody>
</table>

To be anonymous (common to all):
1. Money should not be subject to government tracking.
2. It should be usable 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.

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 reinforcing social ties and location. A definition providing room for sustainable development is needed, eliminating the hierarchical social systems that...
are the product of centrally controlled money, since in most cases this type of money has drawbacks 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 geographical distributed governments. If the idea of modern mutual banking was proposed only in the 19th century (by French social-anarchist Proudhon (Wikipedia, 2006) ), 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 monopolizing 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—envisioned money liberated from interest (Gesell, 1929). The gain of interest for 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 to the community to the users (Van Arkel and Peterse, 1998). He 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 for 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. Gessell 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 Gesells; in some way he is also seeking to use money without the liability of interest; in other ways, there is 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 (Van Arkel and Peterse, 1998).

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 for money. 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. As stated in the title of this paper, 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 future.

The vision of Gesells 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 two and three, in which sustainability is at the core of the socio-economical models to be presented.

We should clarify that the economics of sharing discussed in this paper as one of the phenomena underpinning community currencies are mainly focussed on sharing unused capital and do not include the open source phenomenon or the processes and interactions collectively referred to as the Gift Economy.

### 10.3 Community currencies working framework

Besides the shift in the ideas about money presented above—requiring stronger community participation in the use of money—community currencies can be explained in parallel to the historical evolution of the idea of OCommonsO as a means for structuring rights to access, use and control over resources (Benkler, 2006). Working commons need to define in the first instance their accessibility, either as restricted to members or open to all; in the second instance 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 2.1 below.

<table>
<thead>
<tr>
<th>Table 2. Commons as a unit of cohesion for Community Currencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sourced from description by Benkler (2006))</td>
</tr>
</tbody>
</table>

---

Public 216
The idea of the commons can be extended to community currencies as they have traditionally been geographically localized and their economic interconnections tend to be of a social-geographical nature. (Schraven, 2001). 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, simply 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 irreversible collapse (Linton and Yacub, 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 (Linton and Yacub, 2006):

- Community currency aims to at least provide money that exists in sufficient supply,
- Such money is only for use within a community (it recirculates)
- Ideally this money will be created by its users (sourced from within the community).

Take for example a rural community such as the Medieval village discussed in Section 1, and compare it to its counterpart in modern times, a country village. 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 decay (see Figure 2). However, 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 (see Figure 3).

Figure 10.2: Saving money from village is invested in the urban areas. Consequence village falls into decay (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 countryside 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 less purchases 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 financers or managers of this capital might be placed in a situation in which they receive more money that 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. 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 (Van Arkel and Peterse, 1998), 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 (Cohen-Mitchell, 2000). 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 (Cadwell 2000).

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 though mutualism. Most community currencies 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 countryside village 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. What we are seeing now is the emergence of more effective collective action practices that are decentralized but do not rely on either the price system or management structures (Benkler,2006). Community currencies fall into this category since networked environments provide a more effective platform for the actions of non-profit organizations spreading their activities over a wider community.

Lietaer (2001) 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 utilizing multiple 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 within the community members, the formation of closed commons in which there are regulated norms for trading and membership are 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.

10.4 Economics of sharing

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. The continuous presence of community currencies 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 (Granovetter, 2000). As technological developments make society better networked, new ways of social production emerge (Castells, 1996). Even so, post-modern sharing is emerging from certain physical, rivalrous goods and will increase due to advances in technology (Benkler, 2006). Until now this sharing has not been extended to the spheres of social sharing, in which a third mode of organizing economic production, alongside markets and the state (Benkler, 2004) is being built up.

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, are better expressed in 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. What is understood as sharing is the ability to motivate human interaction or sociability 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.

Current economics of sharing models contextually attached to the use of mobile technology are evolving over services platforms that are social networks in digital forms; the driver to share is expressed in several modes, for example:

- Sharing and distribution of electronic tokens between social networks of friends using I-mode enabled phones in Japan (Rheingold, 2002)
- Sharing tokens, or paying for services through mobile services that are centralized, 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 on 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.
10.5 Technology as a mediator in the use of money and Empirical Business Models

As explained in Section 3, technology and in particular the use of the Internet raised the possibilities and expectations of using technology as a replacement for physical money, as mentioned in Section 1, 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 technology mediated include:

LETSystems: http://www.gmlets.u-net.com
This model is probably the best known and most popular from all the ones already established community currencies. LETS was intended to build local currencies into a sustainable social system according to five criteria (Cadwell,2000):

- 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 groupÕs 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 service, including those that often might appear as redundant in modern society. Members search the electronic catalogue until they find goods or service 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 sellerÕs 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 to 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.

Figure 4 illustrates a closed LETS credits system in a small community of six people. In the example, the trade of credits for goods or services is a dual flow between the issuer of credits (Mr Able) and other members of the community (Mr Baker and Ms Zane). The issuer of credits benefits indirectly from the links between the different members of the community, who can in some cases close the circle (but may not necessarily do that, hence the dotted lines in the diagram) for trading.

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 will 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 our goal of keeping money local. This model is a restricted commons to the Ithaca citizens.

Salt Spring Island Dollars: http://saltspring.gulfislands.com/money/welcome.htm
The citizens from 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% redeemable into their national currency; the second problem was how and 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.

Figure 10.4: An illustration model of how the LETS credits can be used within a local community

Strohalm: http://www.strohalm.org/vlcs.html

The Social Trade Organisation, STRO (in Dutch STROhalm), is a Research & 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 the methodologies of STRO is a circuit of exchange between producers and consumers. A 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 are focusing 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): See Lietaer (2006b).

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 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 Reai (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),
November 5, 2008 D2.1.1 Survey on paradigms

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.

Figure 10.5: Saber Complementary Currency Model (from Lietaer, 2006b)

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: 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 on credit based on the text validation. http://www.economist.com/displayStory.cfm?story_id = 8089667

Starbucks coffee card. 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. https://www.starbucks.com/card/default.asp?cookie%5Ftest = 1

The SIMS: Simelons currency. 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. http://thesims2.co.uk/

LindeX Second Life Virtual World Currency. 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. http://secondlife.com/whatis/currency.php

**My gamma.** This community currency does not exchange money as such but knowledge, social networks, friendships and other tokens. It is particularly popular in Asia (now extending its influence to Europe and Africa). The trading is also token based and the value assigned to some of those tokens is subjective and depends upon the demand for them. Exchange rates are variable and determined by the supply-demand relationships. (See http://www20.mygamma.com/rankdisplay.php?country=ukki = 1 or http://en.wikipedia.org/wiki/MyGamma)

The models listed above are based on sharing common resources provided by technological development based on the theoretical socio-economical 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 them 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 reconceptualized to include the new exchange opportunities opened to all people through the technology.

### 10.6 Further research applied to BIONETS

This document 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. 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 efforts of the user 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 indicators 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. 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 (The Economist, 2006). 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 application for the 21st Century (Rushkoff, 2004)

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 document. Special mention can be given to the planned undersound (Bassoli, 2006) experience where there is the opportunity to create a token trade community currency based on music. The core of research and development currently being undertaken by computing and science researchers in the project points towards a significant benefit of the BIONETS architecture design compared with traditional or currently available mobile networks. For example, the T-Node/U-Node separation, evolving applications, inherent user feedback, etc. might represent a wider technology platform providing more—and *built-in*—support for sharing and community currencies. At this stage of the project these are promising avenues for deploying and exploring such models that require further research.

As an introduction to the concepts in the area of money, community currencies, and economics of sharing, this paper has aimed to link these ideas to applications and services already available, as well as to explore some of the benefits and drawbacks of such uses.
10.7 References


OpenMoney (2006). Available at: www.openmoney.org


Chapter 11

Eigenvector based reputation measures

K. Avrachenkov, D. Nemirovsky, K. S. Pham, R. G. Cascella, R. Battiti, M. Brunato

SUMMARY

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 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, 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 personalized 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 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.

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, and in particular the personalized 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
deliverable 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
two categories. The first category is based on asynchronous methods. The second category is based on the aggrega-
tion/decomposition methods. And the third category is based on the personalization methods which use the information
available locally. We survey in detail all the three categories.

11.1 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.

11.1.1 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 informa-
tion, 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

\[
T_{xj} = (1 - w_p)O_{xj} + w_p \sum_d \frac{R_{dj} \cdot C_{xd}}{C_{xd}},
\]

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.

11.1.2 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 for-
mer 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 \( \theta \) for a node to behave well, as shown in
equation (11.2). It is based on prior experience that is constantly updated by recomputing $\alpha$ and $\beta$ as follows: $\alpha = p + 1$ and $\beta = n + 1$, where $p$ and $n$ are the number of positive and negative feedbacks respectively

$$Beta(\theta, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

(11.2)

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 $C_{ui}$ estimated by node $x$, as defined in the following equation

$$R_{xj} = \frac{\sum F_{ij} \cdot C_{ui}}{\sum C_{ui}}.$$  

(11.3)

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 and many minor unimportant transactions and, then, behave dishonestly in a few large transactions. 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 $i$ time units old, then, the weight of that information is computed as $e^{-\gamma}$ where $\gamma$ is the aging constant that denotes the rate at which information ages. Equation (11.4) 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

$$R_{xj} = \frac{\sum F_{ij} e^{-\gamma i}}{\sum e^{-\gamma i}}.$$  

(11.4)

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 (11.5), where $u$ are the intermediate nodes along the path between the source and the destination

$$R_{xj} = \sum_{e \in \text{incoming}(j)} w_e \frac{R_{xu} \cdot R_{uj}}{\sum_{f \in \text{incoming}(j)} R_{uf}},$$  

(11.5)

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.

### 11.1.3 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 personalized 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

\[
G = cP + (1-c)ev,
\]

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 \( \pi \) such that

\[
\pi G = \pi, \quad \pi e = 1. \tag{11.7}
\]

The row vector \( \pi \) satisfying (11.7) is called a PageRank vector, or simply PageRank. There is an important modification of PageRank called TrustRank [10]. TrustRank adopts a special choice of personalized 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 personalized 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

\[
s_{ij} = sat(i,j) - unsat(i,j). \tag{11.8}
\]

By composing matrix \( P \) in the following way

\[
p_{ij} = \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)}, \tag{11.9}
\]

we can now apply the already familiar PageRank scheme. Personalization vector \( v \) for P2P networks is defined in similar way as in TrustRank. There are some peers in a P2P networks that are know 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 personalized 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]:

\[
\pi = \pi cP + (1-c)v. \tag{11.10}
\]

The structure of the chapter is organized 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 personalized approach to the graph based reputation measures. The personalized 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.

11.2 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 analyze 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 11.1.3, and in particular the personalized 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.

11.3 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 (11.10). Namely,

\[ \pi^{(t+1)} = \pi^{(t)}cP + \frac{1-e^r}{n} e^r, \quad t = 0, 1, \ldots, \]  \hspace{1cm} (11.11)

with \( \pi^{(0)} = \frac{1}{n} e^r \). Since the matrix \( cP \) is substochastic, the algorithm converges. Furthermore, its convergence rate is bounded by \( c \) [13]. The number of FLOPS required to achieve the accuracy \( \varepsilon \) is equal to \( \log_{\frac{1}{\varepsilon}} \text{nnz}(P) \), 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 (11.10) was proposed in [6]. The class of asynchronous iterative methods of [6] can be described as follows:

\[
\pi_{j}^{(t+1)} = \left\{ \begin{array}{ll}
\sum_{i=1}^{n} c \pi_{i}^{(t-d(i,j))} + \frac{1-c}{n} & \text{if } j \in U(t), \\
\pi_{j}^{(t)} & \text{if } j \notin U(t),
\end{array} \right. 
\]

(11.12)

where the function \( U(t) \) gives a set of states to be updated at each step, and the function \( d(i,j) \) 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 11.3.1** Let the functions \( U(t) \) and \( d(i,j) \) 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 \( (t - d(i,j)) \to \infty \) as \( t \to \infty \) as well as \( \forall t : d(i,j) \leq t \).

Then, if the spectral radius of \( cP \) is strictly less than one, every sequence of iterates within the class given by (11.12) 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 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 \( \pi_{j} \) 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 subgraph (“island”) of the whole graph at a node the random walks can be generated at the node until the random walk leaves the subgraph. 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.

### 11.4 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 \( I^{(i)} \), \( i = 1, \ldots, N \), such that

\[
I^{(1)} = \{1, \ldots, n_{1}\}, \\
I^{(2)} = \{n_{1} + 1, \ldots, n_{1} + n_{2}\}, \\
\vdots \\
I^{(N)} = \{\sum_{i=1}^{N-1} n_{i} + 1, \ldots, \sum_{i=1}^{N} n_{i}\},
\]

(11.14)

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:

\[
P = \begin{pmatrix}
P_{11} & P_{12} & \cdots & P_{1N} \\
P_{21} & P_{22} & \cdots & P_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
P_{N1} & P_{N2} & \cdots & P_{NN}
\end{pmatrix}
\]
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,

\[
G = \begin{pmatrix}
G_{11} & G_{12} & \cdots & G_{1N} \\
G_{21} & G_{22} & \cdots & G_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
G_{N1} & G_{N2} & \cdots & G_{NN}
\end{pmatrix}.
\]  
(11.15)

Following the partitioning of the Google matrix, the Rank vector is partitioned into components:

\[
\pi = (\pi_1, \pi_2, \ldots, \pi_N),
\]  
(11.16)

where \( \pi_i \) is a row vector with \( \text{dim}(\pi_i) = n_i \). 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. \( a_{ij} \rightarrow G_{ij} \). Typically the elements of the matrix \( A \) are formed as \( a_{ij} = \zeta_i G_{ij} e \), where \( \zeta_i \) is a probability distribution vector. We call the vector \( \zeta_i \) 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 (11.6) and the Rank vector (11.7), but some of them can be applied to a general (irreducible or primitive) stochastic matrix and its stationary probability distribution.

### 11.4.1 Block-diagonal case

Let us consider the case when all blocks excluding the diagonal ones are zeroes [1], i.e.

\[
P = \begin{pmatrix}
P_1 & 0 & \cdots & 0 \\
0 & P_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & P_N
\end{pmatrix}.
\]

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 \( \pi_i \) be the Rank vector of \( G_i \),

\[
\pi_i = \pi G_i.
\]

Then the original Rank vector \( \pi \) is expressed by

\[
\pi = \left( \frac{n_1}{n} \pi_1, \frac{n_2}{n} \pi_2, \ldots, \frac{n_N}{n} \pi_N \right).
\]

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.

### 11.4.2 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 11.4.1 (Stochastic complement)** For a given index \( i \), let \( G_i \) denote the principal block submatrix of \( G \) obtained by deleting the \( i \)th row and \( i \)th column of blocks from \( G \), and let \( G_{i\i} \) and \( G_{i-1\i} \) designate

\[
G_{i\i} = (G_{i1} G_{i2} \cdots G_{i,j-1} G_{i,j+1} \cdots G_{iN})
\]
November 5, 2008

D2.1.1 Survey on paradigms

and

\[ G_{s_i} = \begin{pmatrix}
G_{i1} & \cdots & G_{i-1,i} & G_{i+1,i} & \cdots & G_{iN_i}
\end{pmatrix}. \]

That is, \( G_{is} \) is the \( i \)th row of blocks with \( G_{ii} \) removed, and \( G_{si} \) 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

\[ S_i = G_{ii} + G_{is}(I - G_i)^{-1}G_{si}. \]

**Theorem 11.4.1** ([25, Theorem 4.1] Coupling theorem) The Rank vector of the Google matrix \( G \) partitioned as (11.15) is given by

\[ \pi = (\nu_1\sigma_1, \nu_2\sigma_2, \ldots, \nu_N\sigma_N), \]

where \( \sigma_i \) are the unique stationary distribution vector for the stochastic complement

\[ S_i = G_{ii} + G_{is}(I - G_i)^{-1}G_{si} \]

and where

\[ \nu = (\nu_1, \nu_2, \ldots, \nu_N) \]

is the unique stationary distribution vector for the aggregation matrix \( A \) whose entries are defined by

\[ a_{ij} = \sigma_iG_{ij}e. \]

In respect to the scope of the survey, Theorem 11.4.1 is given in application to the Google matrix. For the most general formulation of the theorem an interested reader is referred to [25]. Theorem 11.4.1 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 \( \pi \) of a Google matrix \( G \) in \( k \) iterations.

1. Select a vector \( \pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, \ldots, \pi_N^{(0)}) \) with \( \pi^{(0)}e = 1 \).
2. Do \( k = 0, 1, 2 \ldots \)
   (a) Normalize \( \sigma_i^{(k)} = [\pi_i^{(k)}], i = 1, \ldots, N \).
   (b) Form aggregated matrix \( A^{(k)} \)
       \[ a_{ij} = \sigma_i^{(k)}G_{ij}e. \]
   (c) Determine the stationary distribution \( \nu^{(k)} \) of \( A^{(k)} \)
       \[ \nu^{(k)} = \nu^{(k)}A^{(k)}. \]
   (d) Determine disaggregated vector
       \[ \tilde{\pi}^{(k)} = (\nu_1^{(k)}\sigma_1, \nu_2^{(k)}\sigma_2, \ldots, \nu_N^{(k)}\sigma_N). \]
   (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 \( \nu^{(k)} = \nu \). Therefore, \( \tilde{\pi}^{(k)} = \pi \), and \( \pi^{(k+1)} = \pi \).

For the local convergence of the Algorithm 2 it is required to fulfill one of the conditions:

1. \( G \gg 0 \) and \( G \gg \delta Q \), where \( Q = e\pi \).
2. \( G \gg eb \), where \( b \) is a row vector, \( be = \delta \).
Since the Google matrix satisfies 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 \( \delta_1 \). Denote by \( g_{\text{min}} \) the minimum entry of the matrix \( G \). If \( p_{ij} = 0 \) then \( g_{ij} = g_{\text{min}} \). Hence,

\[
g_{\text{min}} = \frac{1 - c}{n}. \tag{11.17}
\]

The maximum of the elements of the Rank vector \( \pi \) 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 realized if there is no other page referring to it. The minimum entry of the Rank vector is \( \frac{1-c}{\pi} \). Therefore the maximum of one of the element of the Rank vector is equal to

\[
\pi_{\text{max}} = 1 - \frac{1-c}{n}(n-1) = \frac{1+c(n-1)}{n}. \tag{11.18}
\]

Hence, if we find \( \delta_1 \) from the constraint

\[
g_{\text{min}} \geq \delta_1 \pi_{\text{max}}, \tag{11.19}
\]

it ensures that \( G \geq \delta_1 Q \). From the equalities (11.17), (11.18) and (11.19) 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 \( \delta_2 \). From the equalities (11.6) we obtain, that \( G \geq \frac{L+1}{2}e \). The equality can be rewritten as \( G \geq e \left( \frac{L+1}{2} e^T \right) \). Therefore, as the vector \( b \) one can take \( \left( \frac{L+1}{2} e^T \right) \). 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 \( J(\pi) \) 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 \( 1 - \delta_1 = \frac{c}{1+c(n-1)} < 1 \),
2. is less than \( \sqrt[1+c(n-1)]{} < 1 \).

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 \( \sqrt[n-1+c]{} \). 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.

### 11.4.3 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 \( 2 \times 2 \) 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 nonsingular [3]. Hence we can factor \( I - G = LDU \) [25, proof of Theorem 2.3], where

\[
L = \begin{pmatrix}
I & 0 \\
-G_{21}(I-G_{11})^{-1} & I
\end{pmatrix},
\]

\[
D = \begin{pmatrix}
I - G_{11} & 0 \\
0 & I - S_2
\end{pmatrix},
\]

\[
U = \begin{pmatrix}
I & -(I-G_{11})^{-1}G_{12} \\
0 & I
\end{pmatrix}.
\]
and where $S_2$ is a stochastic complement of the block $G_{22}$.

Since the matrix $U$ is nonsingular we have $\pi(I - G) = 0$ if and only if $\pi LD = 0$. Hence

$$\pi_2 S_2 = \pi_2, \quad \pi_1 = \pi_2 G_{21}(I - G_{11})^{-1}, \tag{11.20}$$

which means that $\pi_2$ is a stationary distribution for the matrix $S_2$. The expression (11.20) represents a particular case of Theorem 11.4.1 for the $2 \times 2$ decomposition of the transition matrix [25, Corollary 4.1]. The matrix $S_2$ has unique stationary distribution

$$\sigma_2 S_2 = \pi_2, \quad \sigma_2 e = 1.$$  

And we can find $\pi_2$ as $\pi_2 = \rho \sigma_2$, where the factor $\rho$ is chosen to satisfy the normalization condition $\pi e = 1$.

The component $\pi_1$ and the factor $\rho$ can be expressed as components of the stationary distribution of the aggregated matrix

$$A_1 = \begin{pmatrix} G_{11} & G_{12} e \\ \sigma_2 G_{21} & \sigma_2 e \end{pmatrix}.$$ 

From (11.20), $\pi_2 = \rho \sigma_2$ 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 $\alpha$,

$$\alpha A_2 = \alpha, \quad \alpha e = 1.$$ 

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 $\pi$ 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 $k = 0, 1, 2 \ldots$
   (a) Normalize $\sigma_2^{(k)} = [\pi_2^{(k)}]$.
   (b) Form aggregated matrix $A_1^{(k)}$
   $$A_1^{(k)} = \begin{pmatrix} G_{11} & G_{12} e \\ \sigma_2^{(k)} G_{21} & \sigma_2^{(k)} e \end{pmatrix}.$$ 
   (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
   $$\tilde{\pi}^{(k)} = (\omega_1^{(k)}, \rho^{(k)} \sigma_1^{(k)}).$$ 
   (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

$$\tilde{G} = \begin{pmatrix} 0 & 0 \\ G_{21} (I - G_{11})^{-1} & 0 \end{pmatrix}.$$ 

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.
11.4.4 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 \( \pi \) of the Google matrix \( G \) in \( k \) iterations.

1. Determine local Rank vector for each diagonal block \( P_i \)
   
   (a) Normalize \( P_i \), i.e. \( \overline{P}_i = \frac{(P_i)_{jk}}{(P_i)_{jj}} \).
   
   (b) Form \( G_i, G_i = c\overline{P}_i + (1-c)(1/n)E \).
   
   (c) Approximately determine \( \overline{\pi}_i \)
      
      i. Select a vector \( \overline{\pi}_i^{(0)} \).
      
      ii. Do \( k = 1, 2 \ldots \)

   \[ \overline{\pi}_i^{(k)} = \overline{\pi}_i^{(k-1)}G_i. \]

2. Determine BlockRank
   
   (a) Form aggregated matrix \( A \)
      
      \[ a_{ij} = \overline{\pi}_i P_{ij}e. \]
   
   (b) Form \( B, B = cA + (1-c)(1/n)E \).
   
   (c) Approximately determine \( \beta \)
      
      i. Select a vector \( \beta^{(0)} \).
      
      ii. Do \( k = 1, 2 \ldots \)

   \[ \beta^{(k)} = \beta^{(k-1)}B. \]

3. Determine global Rank vector
   
   (a) Form the vector \( \pi^{(0)} = (\beta_1 \overline{\pi}_1, \beta_2 \overline{\pi}_2, \ldots, \beta_N \overline{\pi}_N) \).
   
   (b) Do \( k = 1, 2 \ldots \)

   \[ \pi^{(k)} = \pi^{(k-1)}G. \]

It was empirically shown that Algorithm 4 is faster than the power iteration method by at least the factor of two [18].

11.4.5 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 \( \pi \) 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 \( \overline{\pi}_1 \)
i. Select a vector $\pi_1^{(0)}$.
ii. Do $k = 1, 2 \ldots$

$$\pi_1^{(k)} = \pi_1^{(k-1)} G_1.$$  

(c) Determine aggregation weights of the second stage

$$\eta = \frac{\pi_1}{\sum_{i=1}^{n} (\pi_1)_i}.$$  

2. The second stage: Aggregate non-dangling pages

(a) Form aggregated matrix $G_2$

$$G_2 = \begin{pmatrix} \eta G_{11} e_{n_1} & \eta G_{12} e_{n_2} \\ e_{n_1} v_{n_1} & e_{n_2} v_{n_2} \end{pmatrix}.$$  

(b) Approximately determine $\pi_2$

i. Select a vector $\pi_2^{(0)}$.
ii. Do $k = 1, 2 \ldots$

$$\pi_2^{(k)} = \pi_2^{(k-1)} G_2.$$  

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:

$$\begin{pmatrix} \pi_2 \end{pmatrix}_i = \frac{\left( \frac{\left( \pi_2^{(2)} \right)_i - \left( \pi_2^{(1)} \right)_i}{\left( \pi_2^{(1)} \right)_i - 2 \left( \pi_2^{(2)} \right)_i + \left( \pi_2^{(1)} \right)_i} \right)^2}{\left( \pi_2^{(3)} \right)_i - 2 \left( \pi_2^{(2)} \right)_i + \left( \pi_2^{(1)} \right)_i}. \quad (11.4.6)$$

### 11.4.6 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. $I^{(1)}$ 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 normalized 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

$$S(\pi) = \begin{pmatrix} S(\pi)_1 & 0 & \ldots & 0 \\ 0 & S(\pi)_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & S(\pi)_N \end{pmatrix},$$

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 (11.15), i.e.

$$G_i = (G_{i1}, G_{i2}, \ldots, G_{iN}).$$

**Algorithm 6** Determine an approximation $\pi^{(k)}$ to the Rank vector $\pi$ 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 \( \tilde{G}_{ii} \), i.e. normalize \( G_{ii} \):

\[
(\tilde{G}_{ii})_{jk} = \frac{(G_{ii})_{jk}}{(G_{ii})_{jj}}
\]

(b) Determine the stationary distribution \( \pi_i^{(0)} \) for \( \tilde{G}_{ii} \).

2. Do \( k = 0, 1, 2, \ldots \)

3. the central node

(a) Normalize \( \sigma_i^{(k)} = [\pi_i^{(k)}] \).

(b) Construct the aggregated matrix \( A^{(k)} \) with

\[
a_{ij} = \sigma_i^{(k)} G_{ij} e.
\]

(c) Determine the stationary distribution \( \nu^{(k)} \) for \( A^{(k)} \).

4. For each super-node \( i \)

(a) Construct an \( (n_i+1) \times (n_i+1) \) extended local transition matrix

\[
s_i^{(k)} = \left( \begin{array}{cc}
\tilde{G}_{ii} e^T & (\mathbf{I} - \tilde{G}_{ii}) \nu^{(k)} S(\pi) \\
\nu^{(k)} \sigma_i^{(k)} & 1 - \nu^{(k)} \alpha_k
\end{array} \right) s^k,
\]

where the scalar \( \alpha^k \) 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 \( \tilde{\pi}_i^{(k)} \)

\[
\tilde{\pi}_i^{(k)} = \frac{1 - \nu^{(k)}}{\beta_i^{(k)} - \omega_i^{(k)}}.
\]

5. The central node

(a) Construct vector \( \tilde{\pi}^{(k)} \)

\[
\tilde{\pi}^{(k)} = \left( \tilde{\pi}_1^{(k)}, \tilde{\pi}_2^{(k)}, \ldots, \tilde{\pi}_N^{(k)} \right).
\]

(b) Normalize \( \pi^{(k)} = [\tilde{\pi}^{(k)}] \).

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 \( O(nz((\bar{L} + \bar{U})S(\pi))) + O(n) \), where \( \bar{L} \) and \( \bar{U} \) are low-triangular and upper-triangular matrices of the matrix \( P \), respectively.

11.4.7 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.
11.5 Personalized 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 “localized” or “personalized” 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 personalized 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 personalization vector. Let $\pi_v$ denote the personalized Rank vector (PRV), corresponding to the personalization vector $v$. It can be computed by solving the equation $\pi_v = \pi v$

\[
\pi = \pi c + (1 - c)v; \\
\pi - \pi c = (1 - c)v; \\
\pi(I - cP) = (1 - c)v.
\]

Since $c$ is different from one, the matrix $(I - cP)$ is invertible and we have

\[
\pi_v = v(1 - c)(I - cP)^{-1}.
\]  

Let $Q = (1 - c)(I - cP)^{-1}$. By letting $v = e_i^T$ we see that $\pi_v(e_i) = Q_i$ – the $i^{th}$ row of $Q$. Thus, rows of $Q$ comprise a complete basis for personalized 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 11.5.1** Given two arbitrary $\pi_1, \pi_2$ PRVs and $v_1, v_2$ are their corresponding personalization vectors. Then, for any constants $\alpha_1, \alpha_2 > 0$ such that $\alpha_1 + \alpha_2 = 1$

\[
\alpha_1 \pi_1 + \alpha_2 \pi_2 = c(\alpha_1 \pi_1 + \alpha_2 \pi_2)P + (1 - c)(\alpha_1 v_1 + \alpha_2 v_2)
\]

For any personalization vector $v$, the corresponding PRV is given by : $vQ$. Unfortunately, approach to use the complete basis for the personalized 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

\[
\pi = wQ.
\]

11.5.1 Scaled Personalization

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 personalization 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 personalization algorithm.

**Specification** Let us consider a set of the personalization 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 personalized Rank vector corresponding to $u_h$ is called a basis hub vector $\pi_h$. If the basis vector for each hub node $h \in H$ is computed and stored, then, by Theorem 11.5.1 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.

$e_i$ has 1 in $i^{th}$ place, and 0 elsewhere.
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

$$r'_p(q) = \sum_{t: p \rightarrow q} P[t](1-c)c^{l(t)}, \quad (11.24)$$

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 $t = <w_1, \ldots, w_k>$, then

$$P[t] = \frac{1}{\prod_{i=1}^{k-1} \text{Outdeg}(w_i)},$$

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 $p, q \in V \setminus H$, $r^H_p(q)$ is defined as a restriction of $r_p(q)$ that considers only tours from $p$ to $q$ that pass through $H$, that is,

$$r^H_p(q) = \sum_{t: p \rightarrow H \rightarrow q} P[t](1-c)c^{l(t)}. \quad (11.25)$$

Intuitively, $r^H_p(q)$ is the influence of $p$ on $q$ through $H$. Obviously, if all paths from $p \sim q$ come through $H$, then $r^H_p(q) = r_p(q)$. For carefully chosen $H$, $r_p(q) - r^H_p(q) = 0$ for many pages $p, q$. The strategy is to take advantage of this property by breaking $r_p$ into components $(r_p - r^H_p)$ and $r^H_p$.

$$\pi_p = r_p = (r_p - r^H_p) + r^H_p. \quad (11.26)$$

The vector $(r_p - r^H_p)$ 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 $r^H_p$ 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 11.5.2 For any $p \in V, H \subseteq V$,

$$r^H_p = \frac{1}{1-c} \sum_{h \in H} (r_p(h) - (1-c)x_p(h))(r_h - r^H_h - (1-c)x_h), \quad (11.27)$$

where $x_h = e_h$. The quantity $(r_h - r^H_h)$ appears on the right hand side of (11.27) 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 (11.26) gives

$$r_p = (r_p - r^H_p) + \frac{1}{1-c} \sum_{h \in H} (r_p(h) - (1-c)x_p(h))[(r_h - r^H_h) - (1-c)x_h]. \quad (11.28)$$

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 $(r_p - r^H_p), p \in H$ 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 11.5.3 For any $p \in V$

$$r_p = \frac{c}{|O(p)|} \sum_{i=1}^{|O(p)|} r_{O_i(p)} + (1-c)x_p \quad (11.29)$$

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-neighbors, but with extra importance given to $p$ itself.
Construction of PRV’s  Let \( u = \alpha_1 p_1 + \ldots + \alpha_z p_z \) be a preference vector, where \( p_i \in H \). Let
\[
r_u(h) = \sum_{i=1}^{z} \alpha_i (r_{p_i}(h) - c x_{p_i}(h)).
\]
Equation (11.30)

Then, the PRV \( \pi \) can be computed as follows:
\[
\pi = \sum_{i=1}^{z} \alpha_i (r_{p_i} - r^H_{p_i}) + \frac{1}{1-c} \sum_{h \in H} r_u(h) [(r_h - r^H_h) - (1-c) x_h].
\]
Equation (11.31)

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.

11.5.2 Relation to the A/D approach

Let us relate the Personalized 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 centered on hosts. So, we can encode the personalization vector in the \( k \)-dimensional space. With this adaptation the local Rank vector will not change for different personalizations. Only the BlockRank depends on the personalizations. Therefore, we only need to recompute the BlockRank for each block-personalization vector. The BlockRank algorithm is able to exploit the graph’s block structure to compute efficiently many of the block-oriented basis vectors.

11.6 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 traveling 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 initialization 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 traveling 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. Personalized approach is a kind of auxiliary approach to all the discussed methods because it chooses personalization vector that can be used in aggregation/disaggregation methods as well as in the asynchronous approach and the Monte Carlo method. The personalized approach allows one to take into account preferences like special services provided by nodes or extremely high quality of common services.
Bibliography


[29] The Open Directory Project www.dmoz.org


Chapter 12

Network Formation Games

Giovanni Neglia

Abstract. 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 (be it a group of friends, the World Wide Web or a business and commerce system). In current Internet the network structure arises from interactions of agents at different levels. Internet Service Providers (ISPs) and different organizations 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.

12.1 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 organizations 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-organizing systems, i.e. systems able to reorganize 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 decentralized 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 decentralization.

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 maximizing 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 multicast overlay for real-time contents. Also the behavior 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. Decentralization makes these issues more significant, because in a decentralized 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. Characterizing the structure and the efficiency of networks arising as equilibria from players interaction is one of the main purpose of such research.

12.2 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 organized as follows. In section 12.3 we introduce the two main approaches in current research on network formation games research adopted from two different different communities. In section 12.4 we illustrate the main concepts and tools used in this research field. Finally in section 12.5 we present papers addressing network formation problems in computer networks.

12.3 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 12.4). 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 decentralizing solution to a problem (see section 12.4). 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.
12.4 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 \( s_i \), where \( s_{ij} = 1 \) indicates his willingness to form a link with player \( j \), \( s_{ij} = 0 \) 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 \( s_{ij} \land s_{ji} = 1 \). Given a strategy profile \( s \), we indicate the network rising from player interaction as \( g(s) \), 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 \(-|s_i|\) for user \( i \) as well as the sum of the costs of reaching all the other players:

\[
-u_i(g(s)) = \alpha|s_i| + \sum_{j=1}^{N} d_{ij}(g(s)),
\]

where \( d_{ij}(g(s)) \) is the shortest-path distance (in terms of hops-count) between nodes \( i \) and \( j \) in the graph \( g(s) \).

12.4.1 Nash Equilibrium

A standard equilibrium concept in Game Theory is Nash equilibrium. With reference to our example above we say that a strategy profile \( s^* \) is a Nash Equilibrium if

\[
\forall i, \forall s_i \in S_i, \quad u_i(g(s^*)) \geq u_i(g(s, s_{-i}^*)),
\]

where \( (s_i, s_{-i}^*) \) indicates a strategy profile \( s' \) such that \( s'_j = s^*_j \forall j \neq i \), and \( s'_i = s^*_i \). 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, \( s^* \) is a \( \epsilon \)-approximate Nash Equilibrium (with \( \epsilon > 0 \)) if

\[
\forall i, \forall s_i \in S_i, \quad u_i(g(s^*)) \geq u_i(g(s, s_{-i}^*)) - \epsilon,
\]

while in case of relative improvement bounds, \( s^* \) is a \( \epsilon \)-approximate Nash Equilibrium (with \( 1 > \epsilon > 0 \)) if

\[
\forall i, \forall s_i \in S_i, \quad u_i(g(s^*)) \geq \epsilon u_i(g(s, s_{-i}^*)).
\]

Approximate Nash equilibria have been advocated as tools to quantify the lack of coordination due to selfish behaviour (section 12.4.4) when Nash equilibria do not exist, or are hard to find.

12.4.2 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 \( s_{ij} \land s_{ji} = 1 \).

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 \( u_i() \). 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

\begin{align*}
& a) \quad \forall ij \in g, \quad u_i(g) \geq u_i(g - ij), \\
& b) \quad \forall ij \not\in g, \text{ if } u_i(g + ij) > u_i(g) \text{ then } u_j(g + ij) < u_j(g).
\end{align*}

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 \( s^* \) is a \textit{pairwise Nash equilibrium} if

\[
\begin{align*}
& a) \quad s^* \text{ is a Nash equilibrium,} \\
& b) \quad \forall ij \notin g, \text{ if } u_i(g + ij) > u_i(g) \text{ then } u_j(g + ij) < u_j(g).
\end{align*}
\]

A network \( g \) is \textit{pairwise Nash stable} if there exists a pairwise Nash equilibrium \( s \) of the game, such that \( g = g(s) \).

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 \textit{pairwise stable with transfers} if

\[
\begin{align*}
& a) \quad \forall ij \in g, \quad u_i(g) + u_j(g) \geq u_i(g - ij) + u_j(g - ij) \\
& b) \quad \forall ij \notin g, \quad u_i(g) + u_j(g) \geq u_i(g + ij) + u_j(g + ij).
\end{align*}
\]

12.4.3 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, \( v : G \to \mathbb{R} \). 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 \textit{component additive} if the value of each network \( g \) is equal to the sum of the values of all the disconnected subgraphs (components) in \( g \); it is \textit{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, \( Y : G \times V \to \mathbb{R}^N \), such that \( \sum Y_i(g, v) = v(g) \) for all \( v \) and \( g \). An allocation rule is \textit{component balanced} if the value of each component is divided among players belonging to that component; it is \textit{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:

\[
v(g) = \sum_{i=1}^{N} u_i(g(s)) = - \sum_{i=1}^{N} \left( c_s + \sum_{j=1}^{N} d_{(e,i)}(g(s)) \right),
\]

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.

12.4.4 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, \( s_{eq} \) with the value of an optimal network \( s_{opt} \), i.e. a network for which \( v(s_{opt}) \) is maximum.

\(^1\)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].

\(^2\)Two remarks. First, here we do not need to restrict ourselves to a specific equilibrium, \( s_{eq} \) could be a Nash stable network or one of the other equilibria introduced in section 12.4.2. Second, with a finite number of players the set of possible networks, \( G \), is finite, so \( v() \) has always a maximum value on \( G \).
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 behaviors.

12.5 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. 12.5.1) and global connection games (Sec. 12.5.2). 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 12.5.3 introduces a few games for some specific overlays for file sharing and multicast.

12.5.1 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. 12.1. Being that link creation does not require coordination, the authors adopt Nash equilibrium as investigation tool. They are able to characterize socially optimal networks for all the possible value of \( \alpha \) 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 equilibria\(^3\) 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 \( \alpha \geq 12n \log n \) 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:

\[
-u_i(g(s)) = \alpha |s_i| + \sum_{j=1}^{N} w_{i,j}d_{i,j}(g(s)).
\]

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 \( j \) \( (d_{i,j}(g(s))) \) 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:

\[
-u_i(g(s)) = \sum_{j=1}^{N} w_{i,j}d_{i,j}(g(s)).
\]

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

\(^3\)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.
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 networks 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. 12.1 is considered in [22], where players can establish directed virtual links, building an overlay $g$ on top of an existing underlay $f$, trying to minimize:

$$-u_i(g(s)) = \alpha |s_i| + \sum_{j=1}^N \frac{d(i,j)(g(s))}{d(i,j)(f)},$$

where $d(i,j)(f)$ is the direct distance between node $i$ and $j$ in the underlay graph and $d(i,j)(g(s))$ 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 minimizing the sum of the stretches, the authors can prove an upper bound for the price of anarchy independent from the specific metric space where players can be located. 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 neighbors is uniquely determined by the transmission range: all the nodes inside the transmission range of a node are its neighbors. 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 $i$ $p(i,\text{emit})$ and the received power at node $j$ $P(j,\text{rec})$ is $P(j,\text{rec}) = K_i d(i,j)^\beta P(j,\text{emit})$, and that the received power needs to exceed a minimum level in order to have a successful transmission. In this case power minimization from user $i$ is equivalent to minimize $r_i$. 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 maximize the difference between the number of reachable nodes and $r_i$. 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.

### 12.5.2 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 nonnegative cost $c_e$. The purpose of each player $i$ ($i = 1, \ldots, N$) 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 minimize 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
of the best-response dynamics. A latency cost, constraints on the number of users at each node. The authors also investigate the speed of convergence (not necessarily paths), edge costs that are non-decreasing concave functions of the number of users and can include the Shapley value 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.

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 \( e_i w_i / \sum_j w_j \). In this case there are instances with no pure-strategy Nash equilibrium. For this reason, similarly to [5] the authors look for \( \alpha \)-approximate Nash equilibria, whose cost is within a \( \beta \) factor from the optimal social cost and in particular they try to investigate how much stability one has to give up (higher \( \alpha \) values) in order to achieve low cost solution (low \( \beta \) values), and vice versa. They identify a possible trade-off between \( \beta \) and \( \alpha \) and show that it is very close to the best possible one. The paper presents also an interesting discussion about alternative approaches to \( \alpha \)-approximate Nash equilibria.

### 12.5.3 Overlay Specific Games

In [30] and in its extension [26] we have proposed a network formation game to model interaction among peers using the 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 heterogenous 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 utilization 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 multicast, peers are organized 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 jitters. 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 analyze different multicast 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.
Bibliography


Index

Ad hoc networks, 245
Ad-hoc networks
   Optics and Electrostatics Models, 191
   Road traffic Engineering, 191
ad-hoc networks, 14
Algorithmic Chemistry, 104
Algorithmic game theory, 246
Allocation rule, 248
Artificial Embryogeny, 97, 107
Artificial Evolution, 99
Artificial Regulatory Networks, 103
Autoconstructive Evolution, 102
bloat, 100
Chemical Computing, 104
Chemical Genetic Programming, 104
Chorus, 105
Computational Evolution, 99

Delay Tolerant Networks, 115
DTN, 14, 115
Dynamic Optimization, 101

Economics of sharing, 219
electrostatics, 14
Embryogeny
   Artificial, 97
   Embryonics, 105
   Epidemic Routing, 14
   epidemic routing, 115
   Epigenetic Programming, 104
   ESS, 45, 47
   Evolutionary Algorithms, 99
   Evolutionary Computation, 99
   Evolutionary Computing, 97, 99
   Evolutionary Computing
      Dynamic Optimization Problems, 101
      indirect encoding, 103
   Evolutionary Games, 13, 45
      Evolutionary Stable Strategies, 45
      Replicator Dynamics, 49
   Evolutionary Stable Strategies, 45, 47
Game
   Algorithmic game theory, 246
   Allocation rule, 248
   Evolutionary, 13, 45
   Hawk and Dove, 47
   Nash equilibrium, 247
   Network design games, 245
   Network formation games, 245
   Pairwise stability, 247
   Price of anarchy, 248
   Price of stability, 248
   Value function, 248
   Gene Expression Programming, 104
   Genetic Algorithm, 100
   Genetic Programming, 100
   Genetic Programming
      bloat, 100
      intron, 100
      parsimony, 100
   Hawk and Dove Game, 47
   intron, 100
   mobile ad-hoc networks, 115
   Nash equilibrium, 247
   Network design games, 245
   Network formation games, 245
   ontogenesis, 98
   Ontogenetic Programming, 103
   ontogeny, 98
   opportunistic communications, 115
   pairwise stability, 247
   Peer-to-Peer networks, 245
   phylogenesis, 98
   phylogeny, 98
   Price of anarchy, 248
   Price of stability, 248
   Push programming language, 102
   Replicator Dynamics, 49
   Reputation
      measure, 225
   Scale-free, 14
   self-modifying code, 102
   self-replicating code, 102
   self-reproducing code, 102
   Sensor networks, 245
   spatial models, 14
   Value function, 248

255
Wireless Sensor Networks
   Optimal Deployment, 195