A Short Survey of Aproximation Algorithms for Combinatorial Optimization under Uncertainty[*]

Orestis A. Telelis

Department of Informatics and Telecommunications,
University of Athens, Hellas (Greece)

June 12, 2006

Abstract

This paper briefly describes three well-established frameworks for handling uncertainty in optimization problems. Our focus is mainly on combinatorial optimization and on the development of approximation algorithms under the discussed frameworks. In particular, we give a brief overview of *Stochastic Programming*, *Robust Optimization*, and *Probabilistic Combinatorial Optimization*, and list approximation results from the wealth of recent literature on combinatorial problems under these disciplines.

1 Introduction

Acquisition and validation of input data is one of the most challenging issues in almost every real-world application of operations research techniques. Although several well established theoretical models exist for problems arising in practical applications, direct application of theoretical developments may be difficult or even impossible due to incompleteness of data or due to their questionable validity. Occasionally one may be asked to produce an optimal operational design even before a complete deterministic picture of input data is provided, but only based on estimations and statistical measures. There are several applications where it might be impossible to obtain a current snapshot of the required information, since this information may be subject to constant high-rate change.

Such a situation may arise when one has to deal with market prices and values of needed products. Consider for example the case where we have to purchase certain components in order to build an infrastructure so as to service a community of customers. We may not be able to decide the exact set of customers, because customers wait to see that an infrastructure exists, and subsequently decide whether they are interested in the offered service or not. When service is requested by a set of customers, we will have to extend the existing infrastructure, so as to suit their needs. There are two deficiencies that have to be dealt with in such a situation:

- **Optimization under Uncertainty:** the pre-built infrastructure must be optimized subject to estimations on the customers' needs.

[*]I owe a great part of my understanding of the subject to collaboration and long discussions with Gerasimos G. Pollatos and Vangelis Th. Paschos, and Vassilis Zissimopoulos. I retain full responsibility of any errors found in this manuscript.

- **Market Trends:** Components needed for extending the infrastructure when the exact set of customers is known may become more expensive in presence of such complete information.

Different optimization frameworks have been proposed by the operations research community for handling these deficiencies. These frameworks constitute a means of structuring uncertainty and taking its existence into account during the optimization process. In this short survey we discuss three well-established approaches and give a short survey of the existing bibliography related to applications on combinatorial optimization problems. Our focus is mostly on the computational perspective as pertains to computational hardness results and approximation algorithms.

In the following sections we give an (incomplete) overview of three frameworks for handling uncertainty in combinatorial optimization, namely *Stochastic Programming*, *Robust Discrete Optimization*, and *Probabilistic Optimization.* Note that *robustness* of the designed solution from both feasibility and cost perspectives in the presence of uncertainty is the main purpose of devising these frameworks during an operational design process.

For exposition purposes, we will assume a general *covering* discrete optimization problem: a universe of requirements U and a family $\mathcal{F} \subseteq 2^U$ of subsets of U are given, each $A \in \mathcal{F}$ having an associated cost $c(A)$. We have to select a minimum total cost subset $\mathcal{S} \subseteq \mathcal{F}$ which covers the requirements, i.e. $\bigcup_{A \in \mathcal{S}} = U$. Given this general problem model we discuss next various structures of uncertainty as modeled under the aforementioned frameworks.

2 Stochastic Programming

Stochastic Programming was introduced in the seminal work of G. W. Dantzig [Dan51], and since then it has grown into an important discipline of operations research, that integrates tools from linear programming, stochastic processes, statistics, and probability. For a detailed introduction into the subject, the reader is referred to [BL97]. For latest news, bibliography, and software related to Stochastic Programming the reader is referred to [sto].

Stochastic Programming aims at structuring uncertainty through usage of probability and statistics. That is, although input data may not be present initially during the design process, it is assumed that we can have a statistical estimation of it, or a probability distribution over the possible input. In this short note we discuss two of the most widely used models of Stochastic Programming, namely the *2-stage model with recourse* and the multi-stage model. These two models can be applied in a variety of settings, from which we present the *explicit scenarios*, the *black box*, and the *independent decisions* settings. Descriptions of the 2-stage and multi-stage models are given under the *explicit scenarios* setting, and the other two settings are discussed next.

2.1 2-Stage and Multi-stage Models

In the 2-stage with recourse model the design process extends into two stages. Uncertainty is present in the first stage, while the actual input data materialize in the second stage. However, we are required to build an initial infrastructure (as discussed in the introduction) in a first stage, and extend this infrastructure appropriately in the second stage. Let us give two illustrative examples using the prototype problem used previously.

The first case we consider is *unstable* market trends. Although we have an initial first-stage estimation of $c^0(A)$ for every $A \in \mathcal{F}$ we do not know for certain how these prices evolve in a second stage. This is often parallelized in the stochastic programming literature with known prices of *today* and uncertain prices of *tomorrow*. However, we are able to know, say by statistical measures of previous experience, that the prices of tomorrow evolve into one of k explicitly given *scenarios* with some probability p_r, $r = 1 \dots k$. This means that each $A \in \mathcal{F}$ will cost $c^r(A)$ in second stage with probability p_r, $r = 1 \dots k$. Now we wish to build an initial infrastructure which can be extended in second stage in such a way so that the expected cost over both stages is minimized.

This situation can be modeled by an integer linear program as follows. Let $x^r(A) \in \{0, 1\}$, $A \in \mathcal{F}$, $r = 0 \dots k$ be decision variables that state whether component $A \in \mathcal{F}$ is purchased in first stage or in some second-stage materialized scenario (as part of first-stage extension). Then we have to solve the following ILP:

$$\min Z = \sum_{A \in \mathcal{F}} c^0(A) x^0(A) + \sum_{r=1}^{k} p_r \sum_{A \in \mathcal{F}} c^r(A) x^r(A)$$

$$\text{so that:} \qquad \sum_{A \in \mathcal{F}: u \in A} (x^0(A) + x^r(A)) \geq 1 \quad \forall u \in U, \quad r = 1 \dots k$$

The single type of constraints of this ILP states that each requirement $u \in U$ is covered by some component purchased either in the first or in the second stage.

Now let us proceed with a different example. Consider the case where we know in first stage that our requirements will be from a universe U but we are uncertain of the exact requirements. We are also aware of the today's costs $c^0(A)$ for every $A \in \mathcal{F}$. For the second stage we know that our requirements will form as a subset $U_r \subseteq U$ out of k possible such subsets with probability p_r, $r = 1 \dots k$, but by the time we know the exact subset U_r the prices of the components in $\mathcal{F}$ will have inflated by a factor $\sigma_r \geq 1$, $r = 1 \dots k$ (because when a requirements set materializes, the whole market will try to make the best out of it). This situation can be formed by the following ILP:

$$\min Z = \sum_{A \in \mathcal{F}} c^0(A) x^0(A) + \sum_{r=1}^{k} p_r \sigma_r \sum_{A \in \mathcal{F}} c^0(A) x^r(A)$$

$$\text{so that:} \qquad \sum_{A \in \mathcal{F}: u \in U_r} (x^0(A) + x^r(A)) \geq 1 \quad \forall u \in U_r, \quad r = 1 \dots k$$

In this ILP there is a single type of constraints per realized subset of requirements, stating that each requirement from each such subset must be covered either in first stage or in second stage for the corresponding subset U_r.

The two presented cases differ in the following sense. While the first encodes robustness from an expected cost view the second one also encodes feasibility aspects (since the exact subset of requirements is not known in advance, there is a set of constraints for each different scenario of requirements). In the recent literature there has been a soft distinction of these two cases, stating that the first encodes *data stochasticity*, while the second encodes *demand stochasticity*. We will also see this distinction in the rest of our discussion.

The multi-stage model is the natural extension of the 2-stage model. In this model each stage is associated with a set of scenarios of input data and a probability distribution defined over this set. Note that if each stage is associated with a set of k scenarios and there are l stages, then there can be k^l possible courses of realizations over the sequence of stages. The objective function that must be optimized is a complicated expectation defined over a tree structure of input data. Dynamic programming techniques are naturally employed over such a structure but with limited success, since the size of the instance can grow exponentially large (see [BL97] for an exposition of such techniques).

2.2 Black Box and Independent Decisions Settings

In the *Black Box* setting neither the scenarios nor their distribution of occurence are given explicitly. The distribution is unknown, but it is assumed that we can sample it. That is we can query for a scenario, and the probability that a specific scenario is returned is its occurence probability. Furthermore, in the case of demand-stochasticity, the sampled scenario is accompanied by a corresponding inflation factor of second-stage costs. This model, apart from being more general than the explicit scenarios model in that it eliminates the need for an explicitly given distribution, it also encodes the possibility of exponentially or infinitely many scenarios.

The *Independent Decisions* setting describes demand-stochasticity in the following manner. In our prototypical covering problem each requirement in the universe U is associated with a probability of occurence independently of all other members of U. The set of second-stage realized requirements is given by a coin-flip for each requirement in U. This model encodes exponentially many scenarios (in fact all subsets of U). A single inflation factor is associated with all possible subsets of requirements to be materialized in second stage, and the probabilities associated with elements of U are explicitly given.

2.3 Recent Approximation Results

Recent years have seen an important development of approximation results for stochastic programming, particularly regarding network design problems. The most important results concern generalized techniques for deriving approximation algorithms for Stochastic Programming problems, even for the more general black box and independent decisions settings. For these more general settings and for network design problems in particular, an interesting technique has developed which we discuss in a separate paragraph.

A recent work of Shmoys and Swamy [SS04] presents an algorithmic scheme for deriving approximation algorithms for the 2-stage demand-stochastic black box version of general covering integer programs. Their main result is a general $2\rho + \epsilon$ approximation algorithm for every $\epsilon > 0$, given a ρ-approximation algorithm for the deterministic problem. Their technique involves using an FPRAS for solving linear programs. Applications of this result are exhibited in [SS04] on stochastic versions of the set cover, vertex cover, facility location, multicut (on trees) and multicommodity flow problems. In [SS05b] the authors consider the multistage stochastic versions of integer covering programs under a bounded number of stages and scenarios revealed by black box setting. Their technique consists of using an FPRAS for solving a linear program along with the *Sampled Average* method. See [SS05a] for a survey of their results.

Among the first works concerning development of specialized approximation algorithms for several problems in a stochastic programming setting is the one of Immorlica et al. [IKMM04]. The authors consider 2-stage stochastic versions of the min-cost flow, bin packing, vertex cover, and Steiner tree problems. They show that stochastic min-cost flow can be solved exactly by means of linear programming. Stochastic bin-packing is shown to be solvable at a value arbitrarily close to the optimal through a PTAS. The stochastic element in its definition was the elements that need to be packed. Stochastic 2-stage vertex cover is considered in the explicit scenarios and independent decisions settings, with edges of the underlying graph being the stochastic elements of the problem's definition. A 4- and a 6.3-approximation algorithm is designed for each case respectively. Finally for the stochastic steiner tree problem, in the explicit scenarios model an $O(\log n)$ approximation algorithm is given. We note that in the stochastic steiner tree problem we are given a root vertex, and each second-stage scenario consists of a subset of vertices requiring connection to the root. Therefore we speak of a *rooted* stochastic steiner tree problem.

In another work by Ravi et al. [RS04], 2-stage stochastic versions of the shortest path, bin packing, facility location, vertex cover, and set cover problems are considered. Two versions of the shortest path problem were considered, one involving stochastic sink (with second-stage inflation of edge costs) and one involving stochastic sink and edge costs. The first case was shown to be $O(1)$-approximable while an $O(\log^2 n \log m)$ approximation was given for the second case. The bin-packing problem considered here differed from the one considered in [IKMM04] in that the stochastic element was the sizes of objects to be packed. An APTAS was given for this case. The stochastic version of facility location involved stochastic client set and second-stage facility costs. An 8-approximation LP-rounding algorithm was given. For vertex cover it was shown that an extension of the well-known primal-dual algorithm yields a 2 approximation factor, for the explicit scenarios model. Finally for the explicit scenarios stochastic set cover with each scenario consisting of a subset of ground elements requiring coverage and stochastic second-stage subset costs, a simple extension of the well-known greedy algorithm for set cover yields an $O(\log n \log k)$ approximation (for k scenarios).

In [GRS04] the rooted stochastic steiner tree problem is considered in the 2-stage model under an explicit scenarios setting, with stochastic terminals set. An LP-rounding algorithm based on the LP-relaxation of the flow formulation for steiner tree yields a 40 approximation factor. The same work also exhibits approximation results through LP-rounding for generic stochastic network design problems in the 2-stage explicit scenarios model. For an LP-rounding algorithm with logarithmic approximation factor regarding the 2-stage stochastic minimum spanning tree problem with stochastic edge costs see [DRS05]. In the same paper it is also shown that the logarithmic ratio is the best possible for stochastic minimum spanning tree.

2.4 Cost Sharing and Boosted Sampling for Stochastic Network Design

Usage of the notion of cost-shares from coalitional game theory has recently yielded a successful technique for development of approximation algorithms for the multi-commodity *Rent-or-Buy* network design problem [GKR03]. This technique was later combined with sampling [GPRS04], thus resulting in the generic *Boosted Sampling* framework for approximating stochastic network design problems. The idea of using cost-shares can be briefly summarized in proving the existence of *efficient* (see [GPRS04] for a definition of efficiency) sharing of the cost of the produced solution among constraints involved in the problem's definition (informally, among clients requiring connection over the network).

The achievements of [GPRS04] include a 3.55 approximation for the rooted 2-stage stochastic steiner tree, a 8.45 approximation for stochastic 2-stage facility location, and an 8 approximation for 2-stage stochastic vertex cover, all of them in the *black box* model. Furthermore, in the independent decisions model an 8 approximation is shown for the stochastic steiner forest (where independent decisions concern pairs of vertices requiring connection), 6 approximation for the stochastic facility location, and a 3 approximation for the stochastic vertex cover problems.

The boosted sampling framework was later taken further by an extension for multi-stage stochastic network design [GPRS05]. Results of this work include steiner tree, facility location, and vertex cover. For k stages the approximation factors achieved are respectively $2k$, $3(2^k - 1)$, and $\frac{2}{3}(4^k - 1)$. In this paper a new condition describing the *efficiency* of cost-shares is introduced, regarding manipulation of multiple stages. Another work based on boosted sampling is due to Gupta and Pál [GP05], regarding a 12.6 approximation algorithm for the *unrooted* 2-stage stochastic steiner tree problem. Further developments on cost-sharing are introduced in this paper, so that the need for a given root could be overcome.

It should be noted that, the cost-sharing technique provided a very deep understanding of the sensitivity of the primal-dual algorithm introduced in [AKR95] for the general steiner forest problem. A recent work by Becchetti et al. [BKLP05] provides more simple and efficient cost-shares with respect to this algorithm, thus yielding improved approximation for the multi-commodity rent-or-buy problem, and in effect for the stochastic steiner forest problem in the independent decisions model.

3 Robust Discrete Optimization

Although as mentioned previously all uncertainty-handling frameworks are targeted to production of robust solutions with respect to optimality and feasibility, a relatively more recent framework called *robust optimization* has been proposed as an alternative to Stochastic Programming in the sense that it avoids certain shortcomings of the latter. These shortcomings are discussed in [KY97] in detail and we summarize them here in the following two points:

1. Stochastic Programming requires some statistic knowledge (in terms of distribution estimations) of the input data, which in many cases may be as difficult to acquire as obtaining the actual input data itself. In any case however, acquisition of statistical estimates may be a very time-consuming and error-prone process in lack of an adequately large sample space.

2. Optimization of expectations is a practice of questionable validity in processes involving only a small number of "trials" of the same design operation, because the benefits of an optimum expected value can only be visible in the long term of a large number of trials, especially if a large variance value is involved. Under this light, and given that several operational research activities are commonly judged by their outcomes, a large deviation from the expected value may be disappointing. Therefore other more conservative objectives should be devised.

Starting from these remarks, the authors of [KY97] propose three different optimization criterions, namely *minmax*, *minmax-regret*, and *minmax relative regret*, while also eliminating the need for knowledge of distributions or statistical estimates. We illustrate these criterions using explicit scenarios in a one stage model of our prototypical covering problem. Suppose that we know that costs of components from $\mathcal{F}$ can realize under one of k scenarios, thus the actual cost of each $A \in \mathcal{F}$ may be one of $c^r(A)$, $r = 1 \dots k$.

Under the *minmax* rule we wish to optimize the following ILP:

$$\min Z = \max_{r=1\dots k} \Big(\sum_{A \in \mathcal{F}} c^r(A) x(A) \Big)$$

$$\text{so that:} \qquad \sum_{A \in \mathcal{F}: u \in A} x(A) \geq 1, \quad \forall u \in U, \quad r = 1 \dots k$$

Note that the max aggregator can be easily removed from the objective function by using a standard linear transformation adding k linear constraints to the ILP. The set of constraints simply states that each requirement in U must be covered regardless of cost scenarios. Although we have considered a single stage of decision in this formulation (following closely the formulations presented in [KY97]), the stated ILP can be easily extended to the 2-stage case, where the first stage cost of $A \in \mathcal{F}$ is $c^0(A)$, while the second-stage cost takes one out of k possible values (scenarios) $c^r(A)$, $r = 1 \dots k$:

$$\min Z = \sum_{A \in \mathcal{F}} c^0(A) x^0(A) + \max_{r=1\dots k} \Big(\sum_{A \in \mathcal{F}} c^r(A) x^r(A) \Big)$$

$$\text{so that:} \qquad \sum_{A \in \mathcal{F}: u \in A} (x^0(A) + x^r(A)) \geq 1, \quad \forall u \in U, \quad r = 1 \dots k$$

Keeping the first formulation in mind, the *minmax regret* criterion would dictate optimization of the following (in the one-stage model considered previously):

$$\min Z = \max_{r=1\dots k} \Big(\sum_{A \in \mathcal{F}} c^r(A) x(A) - Z_r^\star \Big)$$

where $Z_r^\star$ in this expression denotes the optimum value for the r-th scenario, $r = 1 \dots k$. This criterion minimizes the maximum deviation from the optimum solution over all scenarios. Accordingly, the *relative minmax regret* objective function is a natural extension of the *min-regret* criterion:

$$\min Z = \max_{r=1\dots k} \Big(\frac{\sum_{A \in \mathcal{F}} c^r(A) x(A) - Z_r^\star}{Z_r^\star} \Big)$$

All three optimization rules have been exhibited on our prototypical covering problem under *data uncertainty* (that is, unknown costs of components available in $\mathcal{F}$). Extensions to 2-stage or multistage models are quite straightforward. The presented models encode the notion of *data robustness*. Only recently was the concept of *demand robustness* discussed and employed [DGRS05], in the context of some network design problems. Demand robustness is the counterpart of demand stochasticity in Stochastic Programming, differing only in the optimization rule and in absence of distribution usage. The corresponding 2-stage covering problem with stochastic demands is modeled under the demand robustness framework as follows (using the minmax rule):

$$\min Z = \sum_{A \in \mathcal{F}} c^0(A) x^0(A) + \max_{r=1\dots k} \Big(\sigma_r \sum_{A \in \mathcal{F}} c^0(A) x^r(A) \Big)$$

$$\text{so that:} \qquad \sum_{A \in \mathcal{F}: u \in A} (x^0(A) + x^r(A)) \geq 1, \quad \forall u \in U_r, \quad r = 1 \dots k$$

3.1 Recent Approximation Results

The literature on approximation algorithms for robust discrete optimization problems is rather scarce. Some hardness results along with some approximation and exact pseudo-polynomial time algorithms can be found in [KY97]. It is worth noting that several well known polynomial time solvable problems become NP-hard in their data-robust version. Results regarding data-robust versions of assignment, shortest path, minimum spanning tree, resource allocation, scheduling, and knapsack problems are developed in [KY97]. Some polynomial-time solvable cases are also discussed.

In a recent work by Aissi et al. [ABV05] the data-robust versions of shortest path, minimum spanning tree, and knapsack problems are revisited under the cases of bounded number of scenarios (by a constant) and unbounded number of scenarios. The authors examine both cases of these problems under *minmax* and *minmax regret* robust optimization objectives and give hardness and approximation results. In particular they show that all problems accept an FPTAS in the case of bounded number of scenarios under the *minmax* rule. The knapsack problem is inapproximable in all other cases, while the shortest path and minimum spanning tree problems bare an FPTAS for bounded scenarios under the *minmax regret* rule also. It is shown that these two problems are not approximable below 2 in all other cases. NP-hardness results of [KY97] for these problems are strengthened in [ABV05].

Demand-robust versions of network design problems are considered for the first time in [DGRS05]. The demand-robust versions of min-cut, min-multi-cut, steiner tree, vertex cover, and facility location problems with explicitly given scenarios. The techniques used in this work are based on LP-rounding, and yield approximation factors $O(\log k)$ for robust min-cut (where k is the number of scenarios) and an exact algorithm on a tree, $O(\log(lm))$ approximation for robust multi-cut (where we are requested to separate l vertex pairs), 30 for robust steiner tree, 4 for robust vertex cover, and 5 for robust facility location.

Let us note that the demand-robust steiner tree problem studied in [DGRS05] generalizes the demand-robust shortest path problem. In both problems we are given a vertex (which is the source for shortest path and the root for steiner tree), and a set of scenarios (each scenario corresponds to a sink for shortest path, and to a set of clients to be connected to the root for steiner tree). Hence the 30 approximation factor achieved for the robust steiner tree also applies for the robust shortest path. A major result shown in [DGRS05] constitutes of a lemma regarding the structure of 2-stage demand-robust problems, which gives a generic lower bound useful in the development of approximation algorithms.

In a subsequent work [GGR06] combinatorial approximation algorithms were developed for 2-stage demand-robust min-cut, shortest paths, and *hitting set* versions of these problems. In particular a $(1+\sqrt{2})$ approximation factor is achieved for robust min-cut and a 7.1 approximation combinatorial algorithm is designed for demand robust shortest paths. We should note that the authors of [GGR06] make use of the structural lemma derived in [DGRS05] and also exhibit the effect of another technique which provokes a separate treatment of the first and second stages in the design of approximation algorithms for these problems.

4 Probabilistic Combinatorial Optimization

The framework of *Probabilistic Combinatorial Optimization* (PCOP) has developed independently of the other two frameworks, although it shares some intuitive characteristics in common with them. Employment of probability distributions specifying the validity of input data is one aspect common to both PCOP and Stochastic Programming. A loose definition of PCOP problems appears in [Ber88], and specifies that these problems are generalized forms of known combinatorial optimization problems including probabilistic elements in the definition of input. The literature regarding the PCOP framework has been relatively scarce, as opposed to the wealth of works in Stochastic Programming, which has formed a separate discipline of operations research. Revived interest in PCOP and some important contributions are due to [Jai85] and [Ber88].

The PCOP framework as presented in [Ber88] appears to be a 2-stage framework also. In particular one is confronted with a data set in a first stage, which has associated probability distributions of validity. In a second stage the actual data set realizes. However, optimization must be performed given the first-stage data set. Once the actual second-stage input realizes, correctional actions must be taken so that the first-stage solution is adapted to the realized data. Thus PCOP also includes some semi-online characteristics. The definition of the optimization rule is however the most intriguing part of the framework, which distinguishes it from the two previously discussed approaches.

Given a set of first-stage probabilistically valid input data we are required to optimize *a priori* over this data so as to be able to efficiently modify this solution as needed to fit the actually realized second-stage data. This solution has often been called an *a priori* solution to the problem. Let $\mathcal{D}$ denote the first-stage probabilistically valid data set, and $S(\mathcal{D})$ denote the a priori solution on $\mathcal{D}$. The PCOP problem is also defined with respect to a modification strategy $\mathcal{M}$ which is going to modify the *a priori* solution so that it fits the second-stage realized data. Let $\mathcal{M}(S(\mathcal{D}), D)$ be the value of the solution output by $\mathcal{M}$ for an a priori solution $S(\mathcal{D})$ and a materialized valid data set $D \subseteq \mathcal{D}$. The *a priori* optimization rule requires that we obtain an *a priori* solution $S(\mathcal{D})$ over $\mathcal{D}$ such that the expected outcome of the modification strategy is minimized:

$$\min Z = E[\mathcal{D}, S(\mathcal{D}), \mathcal{M}] = \sum_{D \subseteq \mathcal{D}} \Pr[D] \mathcal{M}(S(\mathcal{D}), D) \tag{1}$$

A couple of points should be emphasized under this setting:

- The optimization rule (objective function) of a PCOP problem is defined with respect to the chosen modification strategy $\mathcal{M}$. A different modification strategy entails a different PCOP problem.
- Although the design process of solving a PCOP problem is a kind of 2-stage procedure, the robustness criterion of an optimum expectation pertains only to the second stage outcome.
- Optimizing expression 1 essentially requires probabilistic analysis of the modification strategy $\mathcal{M}$ over a distribution of input data.

We illustrate the framework using out prototypical covering problem. Assume that not all components $A \in \mathcal{F}$ are going to be actually valid. However we have a probabilistic estimation $p(A)$ of the validity of each $A \in \mathcal{F}$. Furthermore, once an *a priori* solution is constructed, suppose that a modification strategy $\mathcal{M}$ is going to modify it so that it fits the second stage data. A possible such strategy would be for example to include a component $A \in \mathcal{F}'$ ($\mathcal{F}'$ being the subset of valid components) which covers each uncovered requirement of U after invalidation of some of our *a priori* choices. In order to assure that the problem is feasible we include in $\mathcal{F}$ for each $u \in U$ a singleton component $A(u)$ covering only u and set this compoent's probability of validity to $p(A(u)) = 1$.

There are two major computational challenges associated with a PCOP problem:

- Obtain a polynomial time computable expression for the objective function.
- Optimize the objective function.

It is clear by the general expression given above, that polynomial time evaluation of the objective function given a modification strategy $\mathcal{M}$ and an *a priori* solution $S(\mathcal{D})$ is not straightforward. The stated expression is a sum over all possible subsets of the probabilistic data set, hence some effort is needed to show that this expression can be evaluated efficiently. Alternatively one may be tempted to prove that it is impossible to compute it unless all cases of the sum are enumerated, in which case the problem is $\#P$-complete.

Optimizing the objective function of a PCOP problem amounts to computing an appropriate a priori solution S, which results (after probabilistic analysis) in a minimum value of the expectation in (1). From the perspective of approximation however, we need to have an "optimum" value against which we can compare the obtained expectation value. Given any subset of realized data $D \subseteq \mathcal{D}$ it is obvious that no modification strategy and a priori solution combination can achieve a value less than the optimum value corresponding to D, denoted with $OPT(D)$. Hence the expectation of optima over all possible subsets is a natural optimum against which we can compare the value of (1):

$$E^{\star}[\mathcal{D}] = \sum_{D \subseteq \mathcal{D}} P(D) OPT(D)$$

This is often referred to as the *re-optimization value*, since it is achievable by a modification strategy which ignores the a priori solution and re-optimizes for the particular subset of data realized in second stage.

4.1 Results on Probabilistic Problems

Several well known combinatorial optimization problems have been treated under the PCOP framework. Regarding network design problems the PCOP framework is applied over an input graph each vertex of which is associated with a probability of survival. An alternative formulation concerns assignment of probabilities to the edges of the graph. The *longest path* problem (which is known to be NP-hard in the deterministic setting) has been treated in [MP99] under both settings. The authors propose a modification strategy for both cases and show that evaluation of the corresponding functionals can be done in polynomial time. They also present an exact algorithm finding the optimum solution for the case of metric graphs. In [MP02] the probabilistic version of the *maximum independent set* problem is considered under the a multitude of proposed modification strategies. Polynomial time computable expressions are derived for the corresponding functionals and approximation results are presented on bipartite graphs. Probabilistic versions of the *minimum coloring* problem are treated in [MP03, CEMP05, MP06] where approximation results are derived for general, bipartite, and split graphs.

Less recent results on PCOP problems include the *minimum spanning tree* problem [Ber88, Ber90] and the *travelling salesman facility location* problem [Ber89]. For introductory material on PCOP the reader is referred to [Ber88, BJO90]. Noticeably no general approximation technique has emerged under the PCOP framework, in contrast to the case of stochastic programming. This is mainly due to the fact that PCOP concerns probabilistic analysis of a modification strategy over a distribution of input data, given an a priori solution. Therefore analysis is always heavily dependent on the characteristics of the modification strategy used, as much as on the structure of the underlying deterministic problem studied under the PCOP framework.

References

[ABV05] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation Complexity of min-max (Regret) Versions of Shortest Path, Spanning Tree, and Knapsack. In *Proceedings of the European Symposium on Algorithms, ESA'05*, pages 862–873, 2005.

[AKR95] A. Agrawal, P. N. Klein, and R. Ravi. When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.

[Ber88] D. Bertsimas. *Probabilistic Combinatorial Optimization*. PhD thesis, 1988.

[Ber89] D. Bertsimas. On probabilistic traveling salesman facility location problems. *Transportation Science*, 3:184–191, 1989.

[Ber90] D. Bertsimas. The probabilistic minimum spanning tree problem. *Networks*, 20:245–275, 1990.

[BJO90] D. Bertsimas, P. Jaillet, and A. Odoni. A priori optimization. *Operations Research*, 38(6):1019–1033, 1990.

[BKLP05] L. Becchetti, J. Koenemann, S. Leonardi, and M. Pál. Sharing the cost more efficiently: improved approximation for multicommodity rent-or-buy. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA'05*, pages 375–384, 2005.

[BL97] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag New York, 1997.

[CEMP05] F. D. Croce, B. Escoffier, C. Murat, and V. Th. Paschos. Probabilistic Coloring of Bipartite and Split Graphs. In *Proceddings of the International Conference on Computational Science and Applications, ICCSA (4)'05*, pages 202–211, 2005.

[Dan51] G. W. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1951.

[DGRS05] K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh. How to Pay, Come What May: Approximation Algorithms for Demand-Robust Covering Problems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science, FOCS'05*, pages 367–378, 2005.

[DRS05] K. Dhamdhere, R. Ravi, and M. Singh. On Two-Stage Stochastic Minimum Spanning Trees. In *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization, IPCO'05*, pages 321–334, 2005.

[GGR06] D.l Golovin, V. Goyal, and R. Ravi. Pay Today for a Rainy Day: Improved Approximation Algorithms for Demand-Robust Min-Cut and Shortest Path Problems. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science, STACS'06*, pages 206–217, 2006.

[GKR03] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the ACM Symposium on Theory of Computing, STOC'03*, pages 365–372, 2003.

[GP05] A. Gupta and M. Pál. Stochastic Steiner Trees Without a Root. In *Proceedings of the International Colloquium on Automata, Languages and Programming, ICALP'05*, pages 1051–1063, 2005.

[GPRS04] A. Gupta, Martin Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the ACM Symposium on Theory of Computing, STOC'04*, pages 417–426, 2004.

[GPRS05] A. Gupta, M. Pál, R. Ravi, and A. Sinha. What About Wednesday? Approximation Algorithms for Multistage Stochastic Optimization. In *Proceedings of the International Workshop on Approximation and Randomized Algorithms, APPROX-RANDOM'05*, pages 86–98, 2005.

[GRS04] A. Gupta, R. Ravi, and A. Sinha. An Edge in Time Saves Nine: LP Rounding Approximation Algorithms for Stochastic Network Design. In *Proceedings of the IEEE Symposium on Foundations of Computer Science, FOCS'04*, pages 218–227, 2004.

[IKMM04] N. Immorlica, D. R. Karger, M. Minkoff, and V. S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA'04*, pages 691–700, 2004.

[Jai85] P. Jaillet. *Probabilistic Traveling Salesman Problems.* PhD thesis, 1985.

[KY97] P. Kouvelis and G. Yu. *Robust Discrete Ooptimization and its Applications.* Kluwer Academic Publishers, 1997.

[MP99] C. Murat and V. Th. Paschos. The probabilistic longest path problem. *Networks*, 33(3):207–219, 1999.

[MP02] C. Murat and V. Th. Paschos. A priori optimization for the probabilistic maximum independent set problem. *Theoretical Computer Science*, 270(1–2):561–590, 2002.

[MP03] C. Murat and V. Th. Paschos. The Probabilistic Minimum Coloring Problem. In *Proceedings of the International Workshop on Graph Theoretic Concepts in Computer Science, WG'03*, pages 346–357, 2003.

[MP06] C. Murat and V. Th. Paschos. On the probabilistic minimum coloring and minimum k-coloring. *Discrete Applied Mathematics*, 154(3):564–586, 2006.

[RS04] R. Ravi and A. Sinha. Hedging Uncertainty: Approximation Algorithms for Stochastic Optimization Problems. In *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization, IPCO'04*, pages 101–115, 2004.

[SS04] D. B. Shmoys and C. Swamy. Stochastic Optimization is (Almost) as easy as Deterministic Optimization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science, FOCS'04*, pages 228–237, 2004.

[SS05a] C. Swamy and D. B. Shmoys. Approximation Algorithms for 2-stage and Multistage Stochastic Optimization. In *Proceedings of the International Conference on Algorithms for Optimization with Incomplete Information*, 2005.

[SS05b] C. Swamy and D. B. Shmoys. Sampling-based Approximation Algorithms for Multi-stage Stochastic Optimization. In *Proceedings of the IEEE Symposium on Foundations of Computer Science, FOCS'05*, pages 357–366, 2005.

[sto] Stochastic programming community home page. *http://stoprog.org*.