On first-order methods in stochastic programming

Dissertation submitted for the degree
Doctor of the Hungarian Academy of Sciences

Csaba I. Fábián

John von Neumann University, Kecskemét

2019.

Foreword

This dissertation covers twenty years of my pursuits in stochastic programming. They started with a collaboration with András Prékopa. He invited me to implement a method of his, and in the course of this collaboration I got acquainted with a discipline I found new and fascinating. (The results of this project were published in the joint paper [61].) During this visit I also got acquainted with an optical fiber manufacturing problem that András was then studying. My PhD dissertation [50] was written about this work, and András was my supervisor. On his advice, I started studying bundle methods from a stochastic programming point of view. (Direct results of this line of research are recounted in Chapter 2.) It was due to András's influence that I gave up a career in IT management for one in the academic sphere, a decision I've never regretted. As I had an informatics background, I went into computational stochastic programming. My results extensively rely on the achievements of András and of members of his school: István Deák, János Mayer and Tamás Szántai.

In the years 2007-2010 I collaborated with Gautam Mitra and his team at Brunel University. As a visiting researcher, I spent a month in each of these years at Brunel. Also went to conferences and workshops with that team. I owe many academic acquaintances to Gautam. It was due to him that I got involved in coordination activities of the Stochastic Programming Society. Since Gautam's retirement, I have been collaborating with Achim Koberstein and his colleagues from Paderborn University and the European University Viadrina.

I owe much to the academic leadership of the John von Neumann University, former Kecskemét College, and to my senior colleagues there. They have always strived to provide the conditions of research. Helping young colleagues at Kecskemét is a pleasant obligation for me. One of our projects involves probabilistic problems. Providentially, we can collaborate with Tamás Szántai.

I'm grateful to colleagues and friends in the Hungarian operations research community for encouragement and advice. Especially, I'm obliged to Aurél Galántai who has from time to time read the current versions of my materials and recommended improvements.

Contents

Chapter 1

Introduction

The importance of making sound decisions in the presence of uncertainty is clear in everyday life. Wodehouse in [182], Chapter 6, advocates 'a wholesome pessimism, which, though it takes the fine edge off whatever triumphs may come to us, has the admirable effect of preventing Fate from working off on us any of those gold bricks, coins with strings attached, and unhatched chickens, at which Ardent Youth snatches with such enthusiasm, to its subsequent disappointment.' Advancing age, Wodehouse points out, brings forth prudence. The discipline of stochastic programming promises a faster track.

According to a widely accepted definition, stochastic programming handles mathematical programming problems where some of the parameters are random variables. The Stochastic Programming Society (which exists as a Technical Section of the Mathematical Optimization Society) defines itself as 'a worldwide group of researchers who are developing models, methods, and theory for decisions under uncertainty'. Our approach involves the characterization and modeling of the distribution of the random parameters, and the measuring of risks. The ensuing options may make model building a challenge.

We get a more complete picture of the field by a survey of the achievements that are generally accepted as landmarks. The Stochastic Programming Society honors twelve outstanding researchers with the title 'Pioneer of Stochastic Programming'. I excerpt from the laudations with a focus on theoretic results and applications.

George Dantzig introduced linear programming under uncertainty, presenting the simple recourse model, the two-stage stochastic program and the multi-stage stochastic program. He developed, with A. Madansky, the first decomposition method to solve two-stage stochastic linear programs. He very early foresaw the importance of sampling methods, and later (with P.W. Glynn and G. Infanger) contributed to the development of them. Dantzig's discoveries were continually motivated by applications. His seminal work has laid the foundation for much of the field of systems engineering and is widely used in network design and component design in

computer, mechanical, electrical engineering. He also developed applications for the oil industry (with P. Wolfe), and in aircraft allocation (with A.R. Ferguson).

Michael Dempster studied the solvability of two-stage stochastic programs and provided a bridge between stochastic programming and related statistical decision problems. He also introduced the use of interval arithmetic to SP. In collaboration with J. Birge, G. Gassman, E. Gunn, A. King, and S. Wallace, he participated in the creation of the SMPS standard, the most widely-used data format for SP instances. For decades, he has worked on SP applications in scheduling, finance, and other areas.

Jitka Dupačová proposed a new decision model for the handling of incomplete distributional information. Her minimax approach was one of the forebears of the recent branch of distributionally robust optimization. She also developed the contamination technique that facilitates stability analysis. She made important contributions to asymptotics (including work with R. Wets), and scenario selection/reduction (including work with W. Römisch, G. Consigli, and S. Wallace). She extensively contributed to applications; in economics and finance (including work with M. Bertocchi), water management (including work with Z. Kos, A. Gaivoronski and T. Szántai), and industrial processes (including work with P. Popela).

Yuri Ermoliev with his students has been the major proponent of the stochastic quasigradient method. This approach enables the approximate solution of difficult SP problems. He suggested (with N. Shor) a stochastic analogue of the subgradient process for solving two-stage stochastic programing problems. He worked (with V. Norkin) on measuring, profiling and managing catastrophic risks. He has also worked on other fields of application: pollution control problems, energy and agriculture modeling.

Peter Kall was one of the principal guiding forces in the development of successive approximation methods. He developed bounding approaches with K. Frauendorfer. He developed and promoted, with J. Mayer, the stochastic programming solver system SLP-IOR that has been provided free for educational purposes.

Willem K. Klein Haneveld's early contributions consist of results on marginal problems, moment problems, and dynamic programming. He introduced integrated chance constraints. He was among the first to study stochastic programs with integer variables, both in theory and applications. He worked on SP applications for pension funds and the gas industry.

Kurt Marti has been one of the major proponents of SP approaches to engineering problems, including those arising in structural design and robotics. He also dealt with approximation and stability issues. He developed new algorithms in semi-stochastic approximation and stochastic quasigradient methods.

András Prékopa introduced probabilistic constraints involving dependent random variables. He developed the theory of logconcave measures, a momentous step in the treatment of probabilistic constraints. Together with I. Deák, J. Mayer and T. Szántai, he developed efficient numerical methods for the solution of probabilistic programming problems. He and his co-workers applied his methodology to water systems and power networks in Hungary, and developed a new inventory model, now known as the Hungarian inventory control model. He also worked on the bounding and approximation of expectations and probabilities in higher dimensional spaces. Applications of this theory include communication and transportation network reliability, the characterization of the distribution function of the length of a critical path in PERT, approximate solution of probabilistic constrained stochastic programming problems, and the calculation of multivariate integrals.

Stephen M. Robinson contributed to the study of error bounds and the continuity of solution sets under data perturbations. These results led to his joint work with R. Wets on the stability of two-stage stochastic programs. He has also worked on sample path optimization, and applied these methods to decision models arising in manufacturing and military applications.

R. Tyrell Rockafellar's fundamental results in convex analysis have been extensively applied in SP. With R. Wets, he studied Lagrange multipliers for nonanticipativity constraints. This prepared the way for their development of the progressive hedging algorithm, in which a Lagrangian relaxation of the nonanticipativity constraints allows the use of deterministic solvers. With S. Uryasev and W.T. Ziemba, he has contributed to the foundation of risk management in finance.

Roger J.B. Wets presented work on model classes and properties of these classes (particularly with D. Walkup), and basic algorithmic approaches for solving these models – particularly the L-shaped method (with R. Van Slyke). His early work on underlying structures (with C. Witzgall) later led to many algorithmic developments and preprocessing procedures. He recognized that nonanticipativity can be expressed as constraints, later leading to the progressive hedging algorithm (with R.T. Rockafellar). He has also examined statistical properties of stochastic optimization problems, generalizing the law of large numbers, justifying the use of sampling in solving stochastic programs. He was instrumental in the development of epi/hypo-convergence, approximations and also sampling. Throughout his career, he has been involved in applications of stochastic programming. His first large, and well-known, application is that of lake eutrophication management in Hungary (with L. Somlyódy).

William T. Ziemba is interested in the applications of SP, in particular to portfolio selection in finance. Since 1983 he has been a futures and equity trader and hedge fund and investment manager. He was also instrumental

in the most successful commercial application of SP to a Japanese insurance company, Yasuda Kasai. He is also noted for his applications of SP to race track betting, energy modelling, sports and lottery investments.

The achievements of András Prékopa and his co-workers have a special importance to us. With Tamás Szántai, they developed a stochastic programming model for the optimal regulation of storage levels, and applied it to the water level regulation of lake Balaton. (In connection with this project, they developed a new multidimensional gamma distribution.) With Tamás Szántai, Tamás Rapcsák and István Zsuffa, they developed models for the optimal design and operation of water storage and flood control reservoir systems. He led a project for MVM (Hungarian Electrical Works) for the optimal daily scheduling of power generation. His co-workers were János Mayer, Beáta Strazicky, István Deák, János Hoffer, Ágoston Németh and Béla Potecz. In another project with Sándor Ganczer, István Deák and Károly Patyi, they applied his STABIL model to the electrical energy sector of the Hungarian economy. They were able to dramatically increase the reliability level of the Hungarian power network without cost increase. With Margit Ziermann, he developed the Hungarian inventory control model whose re-ordering rules allowed a substantial decrease in inventory levels.

The stochastic programming school established by András Prékopa is recognized and appreciated by the SP community world-wide. Each of their projects included the development of novel theoretical results, new models and algorithms as well as the implementation of the appropriate solvers. — In Section 7.2, I'll give an overview of the methods and solvers they developed for the solution of probabilistic constrained problems. These solution methods need an oracle that computes or estimates distribution function values and gradients. An abundant stream of research in this direction has been initiated by the work of Prékopa and his school. In Section 7.3, I'll give an overview of these estimation methods.

I mention a further project, the development of a pavement management system for the allocation of financial resources to maintenance and reconstruction works on Hungary's road network. The pavement management system was based on stochastic models, and was developed by András Bakó, Emil Klafszky, Tamás Szántai and László Gáspár. Though I had not yet been acquainted with stochastic programming at the time, I was involved indirectly. They applied my linear programming solver for the solution of the LP equivalents of their stochastic programming problems.

*

I perceive the following recent developments and trends in the stochastic programming field.

Multi-stage models and solution methods have become a heavy-duty tool, especially in energy planning. Effective solution methods have been known for decades, but theoretical justification of multi-stage models was missing. Just a

decade ago, many of us had doubts whether accuracy in the modelling of the random process is compatible with solvability. Recent results in scenario-tree approximation show they are. Paradoxical features of risk measurement in multistage models have been clarified, and safe frameworks have been established.

A new branch has been developing under the name of distributionally robust optimization. The motivation is that the distribution of the random parameters is rarely known in the explicit form required by classic models. We may not even have enough data for a sampling approach, but may know that the distribution belongs to a certain class. The idea is to focus on the worst distribution belonging to that class. (This is analogous to plain robust optimization but avoids over-simplification resulting in over-conservative decisions.) This new modelling approach is inherently related to risk constraints, through duality.

Equilibrium models have appeared in stochastic programming. This is probably due to the de-regulation of the energy markets in Western economies. Given an environment of regulations, resources and random demands, energy producers are expected to compete, and production decisions are modelled on the principles of game theory. Most interesting is the problem of the regulator, who exercises its (limited) influence to steer investment decisions towards a desirable course.

*

This dissertation focusses on computational aspects. Functions involving expectations, probabilities and risk measures typically occur in stochastic programming problems. This means large amounts of data to be organized, and inaccuracy in function evaluations. Interestingly, similar solution approaches proved effective for very diverse problems; enhanced cutting-plane methods in primal and dual forms. I discuss cutting-plane methods in Chapter 2, and their application to the handling of risk measures in Chapter 3.

Decomposition has originally been proposed to overcome the bottleneck of restricted memory. Memory is no longer a scarce resource (entire databases are handled in-memory on today's computers), but decomposition is still relevant, as I'm going to demonstrate in Chapters 4, 5 and 6.

Probabilities are one of the oldest and most intuitive means of controlling risk (you need not explain the concept of 'chance' to a decision maker.) Probabilistic programming is still in the focus of research. I present an overview of current developments in Chapter 7.

The importance of sampling-based methods have significantly increased in recent years, and further hard problems are being attacked by such methods. In Chapter 8, I recount a randomized method bearing a resemblance to stochastic gradient methods. In Chapter 9, I apply this method in an approximation scheme for handling a difficult constraint. In Chapter 10, I adapt the randomized method of Chapter 8 to probability maximization.

Chapter 2

Cutting-plane methods and enhancements

In this chapter we deal with special solution methods for the unconstrained problem

$$\min \varphi(\boldsymbol{x}) \quad \text{such that} \quad \boldsymbol{x} \in X, \tag{2.1}$$

and the constrained problem

$$\min \varphi(\boldsymbol{x}) \quad \text{such that} \quad \boldsymbol{x} \in X,\ \psi(\boldsymbol{x}) \leq 0, \tag{2.2}$$

where $X \subset \mathbb{R}^n$ is a convex bounded polyhedron with diameter D, and φ, ψ are $\mathbb{R}^n \to \mathbb{R}$ convex functions, both satisfying the Lipschitz condition with the constant Λ. We assume that ψ takes positive values as well as 0.

In our context X is explicitly known. The functions, on the other hand, are not known explicitly, but we have an oracle that returns function values and a subgradients at any given point.

A cutting-plane method is an iterative procedure based on polyhedral models. Suppose we have visited iterates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i$. Having called the oracle at these iterates, we obtained linear supporting functions $l_j(\boldsymbol{x})$ $(1 \leq j \leq i)$, i.e.,

$$l_j(\boldsymbol{x}) \leq \varphi(\boldsymbol{x}) \ \ (\boldsymbol{x} \in \mathbb{R}^n) \quad \text{and} \quad l_j(\boldsymbol{x}_j) = \varphi(\boldsymbol{x}_j). \tag{2.3}$$

The current polyhedral model function is the upper cover of these supporting functions,

$$\varphi_i(\boldsymbol{x}) = \max_{1 \leq j \leq i} l_j(\boldsymbol{x}) \quad (\boldsymbol{x} \in \mathbb{R}^n). \tag{2.4}$$

A cutting-plane model of ψ is also built in a similar manner:

$$\psi_i(\boldsymbol{x}) = \max_{1 \leq j \leq i} l'_j(\boldsymbol{x}_j) \qquad (\boldsymbol{x} \in \mathbb{R}^n), \tag{2.5}$$

where l'_j is a supporting linear function to $\psi(\boldsymbol{x})$ at $\boldsymbol{x}_j$ $(j = 1, \ldots, i)$.

Using these objects, a model problem is constructed:

$$\min \varphi_i(\boldsymbol{x}) \quad \text{such that} \quad \boldsymbol{x} \in X,\ \psi_i(\boldsymbol{x}) \leq 0, \tag{2.6}$$

and the next iterate will be a minimizer of the model problem.

2.1 Historical perspectives

Cutting-plane methods for convex problems were proposed by Kelley [87], and Cheney and Goldstein [22] in 1959-1960. These methods are considered fairly efficient for quite general problem classes. However, according to my knowledge, efficiency estimates only exist for the continuously differentiable, strictly convex case. An overview can be found in Dempster and Merkovsky [36], where a geometrically convergent version is also presented.

Though fairly efficient in general, cutting-plane methods are notorious for zigzagging, a consequence of linear approximation. Moreover, starting up is often cumbersome, and cuts tend to become degenerate by the end of the process.

To dampen zigzagging, a natural idea is centralization. It means maintaining a localization set, i.e., a bounded polyhedron that contains the minimizers of the convex problem. A 'central' point in the polyhedron is then selected as the next iterate.

I sketch the procedure as applied to the unconstrained problem (2.1), and in a form that best suits to forthcoming discussions. After the ith iteration, let the localization polyhedron be

$$P_i = \left\{ (\boldsymbol{x}, \phi) \,\middle|\, \boldsymbol{x} \in X,\ \varphi_i(\boldsymbol{x}) \leq \phi \leq \overline{\phi}_i \right\},$$

where $\overline{\phi}_i$ denotes the best function value known at this stage of the procedure. The next iterate will then be the $\boldsymbol{x}$-component of a center of P_i. Different centers have been proposed and tried. The earliest one was the center of gravity by Levin [100]. Applications of this variant have been hindered by the difficulty of computing the center of gravity.

The center of the largest inscribed ball, due to Elzinga and Moore [47], resulted in a successful variant. A more general version, based on the center of the largest-volume inscribed ellipsoid was proposed by Vaidya [171].

The center most used nowadays is probably the analytic center that was introduced by Sonnevend in [158], and has roots in control theory. A cutting-plane method based on the analytic center was proposed by Sonnevend [159]. The method was further developed by Goffin, Haurie, and Vial [71].

2.2 Regularized cutting-plane methods

The following discussion is focused on the unconstrained problem, and I'll treat the constrained problem as an extension.

Regularized cutting-plane approaches have been developed in the nineteen seventies. I sketch the bundle method of Lemaréchal [98], noting that the method is closely related to the proximal point method of Rockafellar [141]. The idea is to maintain a stability center, that is, to distinguish one of the iterates generated that far. The stability center is updated every time a significantly better iterate was found. Roaming away from the current stability center is penalized. Formally, let $\boldsymbol{x}_i^\circ$ denote the stability center after the ith

iteration. The next iterate $\boldsymbol{x}_{i+1}$ will be an optimal solution of the penalized model problem

$$\min_{\boldsymbol{x}\in X} \left\{ \varphi_i(\boldsymbol{x}) + \frac{\rho_i}{2}\|\boldsymbol{x}-\boldsymbol{x}_i^\circ\|^2 \right\}, \tag{2.7}$$

where $\rho_i > 0$ is a penalty parameter. $\boldsymbol{x}_{i+1}$ is often called test point in this context, and the solution is approximated by the sequence of stability centers. As for the new stability center, let

$$\boldsymbol{x}_{i+1}^\circ = \begin{cases} \boldsymbol{x}_{i+1} & \text{if the new iterate is significantly better than } \boldsymbol{x}_i^\circ, \\ \\ \boldsymbol{x}_i^\circ & \text{otherwise.} \end{cases} \tag{2.8}$$

In the former case, $\boldsymbol{x}_i^\circ \to \boldsymbol{x}_{i+1}^\circ$ is called a null step; in the latter, a descent step.

Two issues need further specification: adjustment of the penalty parameter, and interpretation of the term 'significantly better' in the decision about the stability center. Schramm and Zowe [153] discuss these issues and present convergence statements applying former results of Kiwiel [90].

The level method of Lemaréchal, Nemirovskii and Nesterov [99] uses level sets of the model function to regularize the cutting-plane method. Having performed the ith iteration, let

$$\overline{\phi}_i = \min_{1\le j\le i} \varphi(\boldsymbol{x}_j) \quad \text{and} \quad \underline{\phi}_i = \min_{\boldsymbol{x}\in X} \varphi_i(\boldsymbol{x}). \tag{2.9}$$

These are respective upper and lower bounds for the minimum of the convex problem (2.1). Let $\Delta_i = \overline{\phi}_i - \underline{\phi}_i$ denote the gap between these bounds. (The sequence of the upper bounds being monotone decreasing, and that of the lower bounds monotone increasing, the gap is tightening at each step.) Let us consider the level set

$$X_i = \left\{ \boldsymbol{x} \in X \mid \varphi_i(\boldsymbol{x}) \le \underline{\phi}_i + \lambda\Delta_i \right\} \tag{2.10}$$

where $0 < \lambda < 1$ is a level parameter. The next iterate is computed by projecting $\boldsymbol{x}_i$ onto the level set X_i. That is,

$$\boldsymbol{x}_{i+1} = \arg\min_{\boldsymbol{x}\in X_i} \|\boldsymbol{x}-\boldsymbol{x}_i\|^2, \tag{2.11}$$

where $\|.\|$ means Euclidean norm. Setting $\lambda = 0$ gives the pure cutting-plane method. With non-extremal setting, the level sets stabilize the procedure. (The level parameter needs no adjusting in the course of the procedure. That is in contrast with general bundle methods.)

Definition 1 Critical iterations. *Let us consider a maximal sequence of iterations $\boldsymbol{x}_1 \to \cdots \to \boldsymbol{x}_s$ such that $\Delta_1 \ge \cdots \ge \Delta_s \ge (1-\lambda)\Delta_1$ holds. Maximality of this sequence means that $(1-\lambda)\Delta_1 > \Delta_{s+1}$. Then the iteration $\boldsymbol{x}_s \to \boldsymbol{x}_{s+1}$ is labelled critical. This construction is repeated starting from the index $s+1$. Thus the iterations are grouped into sequences, and the sequences are separated with critical iterations.*

Remark 2 *Let $\Delta^{(i)}$ denote the gap after the ith critical iteration. We have $(1-\lambda)\Delta^{(i)} > \Delta^{(i+1)}$ by definition, and hence $(1-\lambda)^i\Delta^{(1)} > \Delta^{(i+1)}$. The number of critical iterations needed to decrease the gap below ϵ is thus on the order of $\log(1/\epsilon)$.*

Given a sequence $\boldsymbol{x}_t \to \cdots \to \boldsymbol{x}_s$ of non-critical iterations, it turns out that the iterates are attracted towards a point that we can call stability center. (Namely, any point from the non-empty intersection $X_t \cap \cdots \cap X_s$ is a suitable stability center. Hence we can look on the level method as a bundle-type method.) Using these ideas, Lemaréchal, Nemirovskii and Nesterov proved the following efficiency estimate. Let $\epsilon > 0$ be a given stopping tolerance. To obtain a gap smaller than ϵ, it suffices to perform

$$c(\lambda)\ \left(\frac{\Lambda D}{\epsilon}\right)^2 \tag{2.12}$$

iterations, where $c(\lambda)$ is a constant that depends only on λ. Even more important is the following experimental fact, observed by Nemirovski in [112], Section 8.2.2. When solving a problem of dimension n with accuracy ϵ, the level method performs no more than

$$n \ln\left(\frac{V}{\epsilon}\right) \tag{2.13}$$

iterations, where $V = \max_X \varphi - \min_X \varphi$ is the variation of the objective over X, that is obviously over-estimated by ΛD. Nemirovski stresses that this is an experimental fact, not a theorem; but he testifies that it is supported by hundreds of tests.

I illustrate practical efficiency of the level method with two figures taken from our computational study [190] where we adapted the level method to the solution of master problems in a decomposition scheme. Figures 2.1 and 2.2 show the progress of the plain cutting-plane method vs the level method in terms of the gap. The gap is measured on a logarithmic scale.

These figures well represent our findings. Cuts in the plain method tend to become shallow, as in Figure 2.1, while the level method shows a steady progress. Moreover, initial iterations of the plain method are often ineffective, as shown in Figure 2.2. Starting up causes no problem for the level method.

Remark 3 *All the above discussion about the level method and the corresponding results remain valid if the lower bounds $\underline{\phi}_i$ $(i = 1, 2, \ldots)$ in (2.9) are not set to be respective minima of the related model functions, but set more generally, observing the following rules:*

the sequence $\underline{\phi}_i$ is monotone increasing, and

$\min_{\boldsymbol{x}\in X} \varphi_i(\boldsymbol{x}) \ \le\ \underline{\phi}_i \ \le\ \overline{\phi}_i$ *holds for every i.*

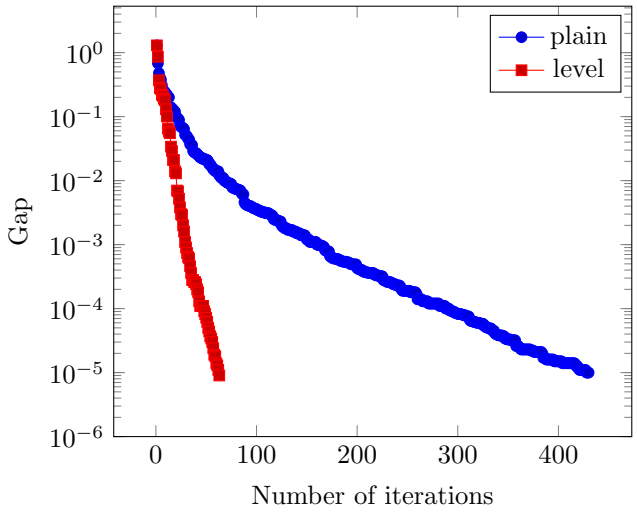

Figure 2.1: Decrease of the gap: plain cutting-plane method vs level method

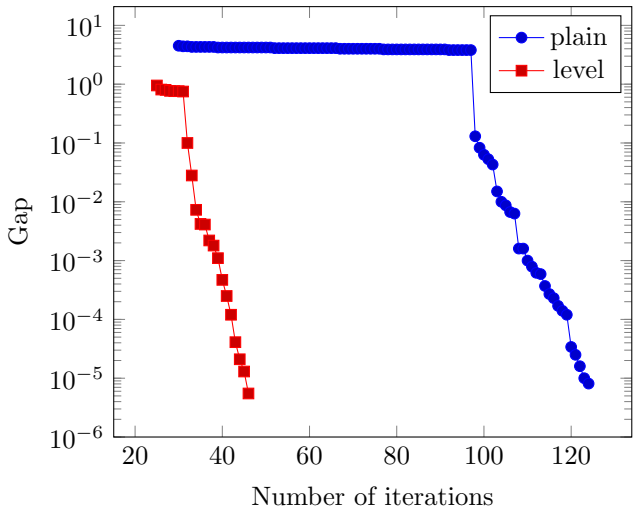

Figure 2.2: Decrease of the gap: plain cutting-plane method vs level method

Lemaréchal, Nemirovskii and Nesterov extended the level method to the solution of the constrained problem (2.2). Their constrained level method is a primal-dual method, where the dual variable $\alpha \in \mathbb{R}$ is kept unchanged as long as possible. The procedure consists of runs of an unconstrained method applied to the composite objective $\alpha\varphi(\boldsymbol{x}) + (1-\alpha)\psi(\boldsymbol{x})$. – To be precise, we speak of runs of a special unconstrained method that satisfies the criteria of Remark 3.

Let Φ denote the optimal objective value of problem (2.2). If Φ is known in advance, then the quality of an approximate solution $\boldsymbol{x} \in X$ can be measured by $\max\{\, \varphi(\boldsymbol{x}) - \Phi,\, \psi(\boldsymbol{x}) \,\}$.

Let moreover $\underline{\Phi}_i$ denote the optimal objective value of the model problem (2.6). This is a lower approximation for Φ.

The best point after iteration i is constructed in the form of a convex combination of the former iterates:

$$\boldsymbol{x}_i^\star = \sum_{j=1}^{i} \varrho_j \boldsymbol{x}_j, \tag{2.14}$$

where the weights are determined through the solution of the following problem

$$\begin{aligned} \min \quad & \max\left\{ \sum_{j=1}^{i} \varrho_j \varphi(\boldsymbol{x}_j) - \underline{\Phi}_i,\; \sum_{j=1}^{i} \varrho_j \psi(\boldsymbol{x}_j) \right\} \\ \text{such that} \quad & \varrho_j \geq 0 \;(j = 1, \ldots, i), \quad \sum_{j=1}^{i} \varrho_j = 1. \end{aligned} \tag{2.15}$$

The linear programming dual of (2.15) is written as $\max_{\alpha\in[0,1]} h_i(\alpha)$ with

$$h_i(\alpha) = \min_{1\leq j\leq i} \left\{\, \alpha(\varphi(\boldsymbol{x}_j) - \underline{\Phi}_i) + (1-\alpha)\psi(\boldsymbol{x}_j) \,\right\}. \tag{2.16}$$

The next dual iterate α_i is set according to the following construction. Let $I_i \subseteq [0,1]$ denote the interval over which $h_i(\alpha)$ takes non-negative values. Let moreover the subinterval $\hat{I}_i \subset I_i$ be obtained by shrinking I_i: the center of $\hat{I}_i$ is the same as the center of I_i, and for the lengths, $|\hat{I}_i| = (1-\mu)|I_i|$ holds with some preset parameter $0 < \mu < 1$. The rule is then to set

$$\alpha_i = \begin{cases} \alpha_{i-1} & \text{if} \quad i > 1 \quad \text{and} \quad \alpha_{i-1} \in \hat{I}_i, \\ \text{center of } I_i & \text{otherwise.} \end{cases} \tag{2.17}$$

The next primal iterate $\boldsymbol{x}_{i+1}$ is selected by applying a level method iteration to the composite objective function $\alpha_i\varphi(\boldsymbol{x}) + (1-\alpha_i)\psi(\boldsymbol{x})$, with the cutting-plane model $\alpha_i\varphi_i(\boldsymbol{x}) + (1-\alpha_i)\psi_i(\boldsymbol{x})$. The best function value taken among the known iterates is $\bar{\phi}_i = \alpha_i\underline{\Phi}_i + h_i(\alpha_i)$. A lower function level is selected specially as

$$\underline{\phi}_i = \alpha_i\underline{\Phi}_i. \tag{2.18}$$

Using these objects and ideas, Lemaréchal, Nemirovskii and Nesterov proved the following efficiency estimate. Let $\epsilon > 0$ be a given stopping tolerance. To

obtain an ϵ-optimal ϵ-feasible solution, it suffices to perform

$$c(\mu,\lambda)\ \left(\frac{2\Lambda D}{\epsilon}\right)^2\ \ln\left(\frac{2\Lambda D}{\epsilon}\right) \tag{2.19}$$

iterations, where $c(\mu,\lambda)$ is a constant that depends only on the parameters. The idea of the proof is to divide the iterations into subsequences in whose course the dual iterate does not change. The proof is based on the following propositions.

Proposition 4 *Assume that $\epsilon > \max_{\alpha\in[0,1]} h_i(\alpha)$ holds.*

Then $\boldsymbol{x}_i^\star$ is an ϵ-feasible ϵ-optimal solution for the constrained problem (2.2), i.e., $\boldsymbol{x}_i^\star \in X$, $\psi(\boldsymbol{x}_i^\star) \le \epsilon$ and $\varphi(\boldsymbol{x}_i^\star) \le \Phi + \epsilon$.

Proposition 5 *$h_i(\alpha_i) \ge \frac{\mu}{2}\max_{\alpha\in[0,1]} h_i(\alpha)$ always holds with the dual iterate selected according to (2.17).*

Proposition 6 *Consider a sequence of iterations in the course of which the dual iterate does not change; namely, let $t < s$ be such that $\alpha_t = \cdots = \alpha_s$.*

If $s - t > c(\lambda)\left(\frac{\Lambda D}{\varepsilon}\right)^2$ holds with some $\varepsilon > 0$, then $h_s(\alpha_s) \le \varepsilon$ follows. – Here $c(\lambda)$ is the constant of the efficiency estimate (2.12).

Proposition 7 *Let $1 < t < s$ be such that $\alpha_{t-1} \ne \alpha_t = \cdots = \alpha_{s-1} \ne \alpha_s$.*

Then $|I_t| \ge (2-\mu)|I_s|$ holds due to the selection rule of the dual iterate.

The key is Proposition 6 that is a consequence of the efficiency estimate (2.12) of the level method. The specially selected lower levels (2.18) satisfy the rules of Remark 3.

2.3 Working with inexact data

Exact supporting functions of the form (2.3) are often impossible to construct, or may require an excessive computational effort. A natural idea is to construct approximate supporting functions. Given the current iterate $\hat{\boldsymbol{x}}$ and the accuracy tolerance $\hat{\delta} > 0$, let the oracle return a linear function satisfying

$$\ell(\boldsymbol{x}) \le \varphi(\boldsymbol{x})\ \ (\boldsymbol{x}\in\mathbb{R}^n) \quad \text{and} \quad \ell(\hat{\boldsymbol{x}}) \ge \varphi(\hat{\boldsymbol{x}}) - \hat{\delta}. \tag{2.20}$$

The simplest idea is to construct a sequence $\delta_i \searrow 0$ before optimization starts. In the course of the ith oracle call, the accuracy tolerance $\hat{\delta} = \delta_i$ will then be prescribed. Zakeri, Philpott and Ryan [187] applied this approach to the pure cutting-plane method, others to bundle methods.

Kiwiel [89] developed an inexact bundle method, where 'the accuracy tolerance is automatically reduced as the method proceeds. The reduction is, on the one hand, slow enough to save work by allowing inexact evaluations far from a solution, and, on the other hand, sufficiently fast to ensure that the method generates a minimizing sequence of points.' At each iteration, Kiwiel estimates the quantity by which the new test point can improve on the current stability

center. He regulates accuracy by halving the oracle tolerance every time it is found too large as compared to this estimate of potential improvement.

In [51], I developed an approximate version of the level method. The idea was to always set the accuracy tolerance in proportion to the current gap. — The method was called 'inexact' in the paper. In view of future developments, though, I'm going to call it 'approximate' in this dissertation. — I extended the convergence proof of Lemaréchal, Nemirovskii and Nesterov [99] to the approximate level method. Subsequent computational studies ([62], [184]) indicate that my approximate level method inherits the experimental efficiency (2.13). I also worked out an approximate version of the constrained level method, and extended the efficiency estimate (2.19) to the approximate version.

Kiwiel [91] developed a partially inexact version of the bundle method. Together with a new test point $\boldsymbol{x}_{i+1}$, a descent target $\phi_{i+1} \in \mathbb{R}$ is always computed, based on the predicted objective decrease that occurs when moving from $\boldsymbol{x}_i^\circ$ to $\boldsymbol{x}_{i+1}$. A descent step is taken if $\varphi(\boldsymbol{x}_{i+1}) \leq \phi_{i+1}$, a null step otherwise. The oracle works as follows. Passed the current test point $\hat{\boldsymbol{x}}$ and descent target $\hat{\phi}$, it returns a linear function $\ell(\boldsymbol{x}) \leq \varphi(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^n)$ such that

$$\begin{array}{lll} \text{either} & \ell(\hat{\boldsymbol{x}}) > \hat{\phi}, & \text{certifying that the descent target cannot be attained,} \\ \text{or} & \ell(\hat{\boldsymbol{x}}) \leq \hat{\phi}, & \text{in which case } \ell(\hat{\boldsymbol{x}}) = \varphi(\hat{\boldsymbol{x}}) \text{ should hold.} \end{array} \tag{2.21}$$

No effort is devoted to approximating the function at the new test point, if no significant improvement can be expected.

De Oliveira and Sagastizábal [27] developed a general approach for the handling of inaccuracy in bundle and level methods. Their analysis 'considers novel cases and covers two particular on-demand accuracy oracles that are already known: the asymptotically exact oracles from [51, 62, 187]; and the partially inexact oracle in [91].' — In the taxonomy of de Oliveira and Sagastizábal, methods that drive the accuracy tolerance to 0 are called 'asymptotically exact'. I'm going to call them 'approximate' in this dissertation, in accord with the terminology introduced above.

[27] also contains a thorough computational study. The authors solved two-stage stochastic programming problems from the collection of Deák, described in [33]. A decomposition framework was implemented and the master problems were solved with cutting-plane, bundle and level methods, applying different approaches for the handling of inexact data. Regularized methods performed better than pure cutting-plane methods. Among regularized methods, level methods performed best. Inexact function evaluations proved generally effective. The best means of handling inexact data proved to be a combination of my approximate level method and of Kiwiel's partially inexact approach. ([27] won Charles Broyden Prize, awarded annually to the best paper published in the Optimization Methods and Software journal.)

2.4 Recent contribution

In [64] and [184] I worked out a special version of the on-demand accuracy approach of de Oliveira and Sagastizábal [27]. — According to the taxonomy of [27], my method falls into the 'partly asymptotically exact' category, and this term was used also in our papers [64] and [184]. In this dissertation, I'm going to call the method 'partially inexact' to keep the terminology simple. (The latter term is in accord with Kiwiel's terminology of [91].)

My specific version is interesting for two reasons. First, it enables the extension of the on-demand accuracy approach to constrained problems. Second, the method admits a special formulation of the descent target (specified in Proposition 11, below). This formulation indicates that the method combines the advantages of the disaggregate and the aggregate models when applied to two-stage stochastic programming problems. (This will be discussed in Chapter 4.)

In the following description of the partially inexact level method, the iterations where the descent target has been attained are called substantial. $\mathcal{J}_i \subset \{1, \ldots, i\}$ denotes the set of the indices belonging to substantial iterates up to the ith iteration. If the jth iteration is substantial then the accuracy tolerance δ_j is observed in the corresponding approximate supporting function. Formally, $l_j(\boldsymbol{x}_j) + \delta_j \geq \varphi(\boldsymbol{x}_j)$ holds for $j \in \mathcal{J}_i$. The best upper estimate for function values up to iteration i is

$$\overline{\phi}_i = \min_{j \in \mathcal{J}_i} \{\, l_j(\boldsymbol{x}_j) + \delta_j \,\}. \tag{2.22}$$

The accuracy tolerance is always set to be proportional to the current gap, i.e., we have $\delta_{i+1} = \gamma \Delta_i$ with an accuracy regulating parameter γ $(0 < \gamma \ll 1)$.

Algorithm 8 *A partially inexact level method.*

8.0 Parameter setting.

Set the stopping tolerance $\epsilon > 0$.

Set the level parameter λ $(0 < \lambda < 1)$.

Set the accuracy regulating parameter γ such that $0 < \gamma < (1 - \lambda)^2$.

8.1 Bundle initialization.

Let $i = 1$ (iteration counter).

Find a starting point $\boldsymbol{x}_1 \in X$.

Let $l_1(\boldsymbol{x})$ be a linear support function to $\varphi(\boldsymbol{x})$ at $\boldsymbol{x}_1$.

Let $\delta_1 = 0$ (meaning that l_1 is an exact support function).

Let $\mathcal{J}_1 = \{1\}$ (set of substantial indices).

8.2 Model construction and near-optimality check.

Let $\varphi_i(\boldsymbol{x}) = \max_{1 \leq j \leq i} l_j(\boldsymbol{x})$ be the current model function.

Compute $\underline{\phi}_i = \min_{\boldsymbol{x} \in X} \varphi_i(\boldsymbol{x})$, and let $\overline{\phi}_i = \min_{j \in \mathcal{J}_i} \{l_j(\boldsymbol{x}_j) + \delta_j\}$.

Let $\Delta_i = \overline{\phi}_i - \underline{\phi}_i$. If $\Delta_i < \epsilon$ then near-optimal solution found, stop.

8.3 Finding a new iterate.

Let $X_i = \left\{ \boldsymbol{x} \in X \,|\, \varphi_i(\boldsymbol{x}) \leq \underline{\phi}_i + \lambda\Delta_i \right\}$.

Let $\boldsymbol{x}_{i+1} = \arg\min_{\boldsymbol{x}\in X_i} \|\boldsymbol{x} - \boldsymbol{x}_i\|^2$.

8.4 Bundle update.

Let $\delta_{i+1} = \gamma\Delta_i$.

Call the oracle with the following inputs:
- the current iterate $\boldsymbol{x}_{i+1}$,
- the accuracy tolerance δ_{i+1}, and
- the descent target $\overline{\phi}_i - \delta_{i+1}$.

Let $l_{i+1}(\boldsymbol{x})$ be the linear function returned by the oracle.

If the descent target was reached then let $\mathcal{J}_{i+1} = \mathcal{J}_i \cup \{i+1\}$, otherwise let $\mathcal{J}_{i+1} = \mathcal{J}_i$.

Increment i, and repeat from step *8.2*.

Specification 9 *Oracle for Algorithm 8.*

The input parameters are
$\hat{\boldsymbol{x}}$: the current iterate,
$\hat{\delta}$: the accuracy tolerance, and
$\hat{\phi}$: the descent target.

The oracle returns a linear function $\ell(\boldsymbol{x})$ such that

$$\ell(\boldsymbol{x}) \leq \varphi(\boldsymbol{x}) \quad (\boldsymbol{x} \in \mathbb{R}^n), \qquad \|\nabla\ell\| \leq \Lambda, \qquad \text{and}$$

either $\ell(\hat{\boldsymbol{x}}) > \hat{\phi}$, certifying that the descent target cannot be attained,
or $\ell(\hat{\boldsymbol{x}}) \leq \hat{\phi}$, in which case $\ell(\hat{\boldsymbol{x}}) \geq \varphi(\hat{\boldsymbol{x}}) - \hat{\delta}$ should also hold.

Theorem 10 *To obtain $\Delta_i < \epsilon$, it suffices to perform $c(\lambda,\gamma)\left(\frac{\Lambda D}{\epsilon}\right)^2$ iterations, where $c(\lambda,\gamma)$ is a constant that depends only on λ and γ.*

Proof. This theorem is a special case of Theorem 3.9 in de Oliveira and Sagastizábal [27]. – The key idea of the proof is that given a sequence $\boldsymbol{x}_t \to \cdots \to \boldsymbol{x}_s$ of non-critical iterations according to Definition 1, an upper bound can be given on the length of this sequence, as a function of the last gap Δ_s. □

A simpler proof can be composed by extending the convergence proof of the approximate level method in Fábián [51]. Theorem 7 in [51] actually applies word for word, only (2.22) needs to be substituted for the upper bound. I abstain from including this proof.

The computational study of [184] indicates that the partially inexact level method inherits the experimental efficiency (2.13).

The partially inexact level method admits a special formulation of the descent target. Let

$$\kappa = \frac{\gamma}{1-\lambda} \tag{2.23}$$

with the parameters λ, γ set in step *8.0* of Algorithm 8. Of course we have $0 < \kappa < 1$.

Proposition 11 *The efficiency estimate of Theorem 10 remains valid with the descent target $\kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i$ set in step 8.4 of the partially inexact level method.*

Proof. Let us first consider the case $i > 1$ and the iteration $\boldsymbol{x}_{i-1} \to \boldsymbol{x}_i$ was non-critical according to Definition 1. We are going to show that the descent target remains unchanged in this case, i.e.,

$$\kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i = \overline{\phi}_i - \delta_{i+1}. \tag{2.24}$$

Due to the non-criticality assumption we have $(1-\lambda)\Delta_{i-1} \le \Delta_i$. Hence by the definition of δ_i and the parameter setting $\gamma < (1-\lambda)^2$ we get

$$\delta_i = \gamma\Delta_{i-1} \le \frac{\gamma}{1-\lambda}\Delta_i < (1-\lambda)\Delta_i. \tag{2.25}$$

Let us observe that

$$\varphi_i(\boldsymbol{x}_i) + \delta_i \ge \overline{\phi}_i \tag{2.26}$$

holds, irrespective of $\boldsymbol{x}_i$ being substantial or not. (In case $i \in \mathcal{J}_i$, this follows from the definition of $\overline{\phi}_i$; otherwise, a consequence of $\overline{\phi}_i = \overline{\phi}_{i-1}$.)

From (2.26) and (2.25) follows

$$\varphi_i(\boldsymbol{x}_i) \ge \overline{\phi}_i - \delta_i > \overline{\phi}_i - (1-\lambda)\Delta_i = \underline{\phi}_i + \lambda\Delta_i. \tag{2.27}$$

(The equality is a consequence of $\Delta_i = \overline{\phi}_i - \underline{\phi}_i$.)

The new iterate $\boldsymbol{x}_{i+1}$ found in step *8.3* belongs to the level set X_i, hence we have

$$\varphi_i(\boldsymbol{x}_{i+1}) \le \underline{\phi}_i + \lambda\Delta_i. \tag{2.28}$$

The function $\varphi_i(\boldsymbol{x})$ is continuous, hence due to (2.27) and (2.28) there exists $\widehat{\boldsymbol{x}} \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$ such that $\varphi_i(\widehat{\boldsymbol{x}}) = \underline{\phi}_i + \lambda\Delta_i$. We are going to show that equality holds in (2.28). The assumption $\varphi_i(\boldsymbol{x}_{i+1}) < \underline{\phi}_i + \lambda\Delta_i$ leads to a contradiction, because in this case $\widehat{\boldsymbol{x}} \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$ should hold, implying $\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}\|^2 < \|\boldsymbol{x}_i - \boldsymbol{x}_{i+1}\|^2$. Obviously $\widehat{\boldsymbol{x}} \in X_i$, which contradicts the definition of $\boldsymbol{x}_{i+1}$.

Hence we have equality in (2.28). From this and the selection of κ we obtain

$$\kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i \;=\; \kappa\left(\underline{\phi}_i + \lambda\Delta_i\right) + (1-\kappa)\overline{\phi}_i \;=\; \overline{\phi}_i - \kappa(1-\lambda)\Delta_i,$$

which proves (2.24) due to the setting $\kappa = \frac{\gamma}{1-\lambda}$.

Let us now consider the case when the iteration $\boldsymbol{x}_{i-1} \to \boldsymbol{x}_i$ was critical. The upper bound mentioned in the proof of Theorem 10 applies to the sequence of

non-critical iterations just preceding $\boldsymbol{x}_{i-1} \to \boldsymbol{x}_i$. Hence the same estimate applies to the total number of non-critical iterations. (The linear function $l_{i+1}(\boldsymbol{x})$ generated by the modified descent target may prove useless, resulting in an extraneous iteration. However, the number of critical iterations is small – on the order of $\log(1/\epsilon)$ as noted in Remark 2.) □

An analogue of Remark 3 holds for the partially inexact level method:

Remark 12 *All the above discussion about the partially inexact level method and the corresponding results remain valid if the lower bounds $\underline{\phi}_i$ $(i = 1, 2, \ldots)$ are not set to be respective minima of the related model functions, but set more generally, observing the following rules: the sequence $\underline{\phi}_i$ is monotone increasing ; $\underline{\phi}_i \leq \overline{\phi}_i$ holds ; and $\underline{\phi}_i$ is not below the minimum of the corresponding model function over X.*

In [64], I extended the on-demand accuracy approach to constrained problems. Let $l_j(\boldsymbol{x})$ and $l'_j(\boldsymbol{x})$ denote the approximate support functions constructed to $\varphi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$, respectively, in iteration j. Like in the unconstrained case, we distinguish between substantial and non-substantial iterates. Let $\mathcal{J}_i \subset \{1, \ldots, i\}$ denote the set of the indices belonging to substantial iterates up to the ith iteration. If $j \in \mathcal{J}_i$ then we have $l_j(\boldsymbol{x}_j) + \delta_j \geq \varphi(\boldsymbol{x}_j)$ and $l'_j(\boldsymbol{x}_j) + \delta_j \geq \psi(\boldsymbol{x}_j)$, with a tolerance δ_j determined in the course of the procedure.

The best point $\boldsymbol{x}_i^\star$ after iteration i is constructed as a convex combination of the iterates $\boldsymbol{x}_j$ $(j \in \mathcal{J}_i)$. The weights ϱ_j $(j \in \mathcal{J}_i)$ are determined through the solution of the following problem:

$$\begin{aligned} \min \quad & \max \left\{ \sum_{j \in \mathcal{J}_i} \varrho_j \left(l_j(\boldsymbol{x}_j) + \delta_j \right) - \underline{\Phi}_i, \sum_{j \in \mathcal{J}_i} \varrho_j \left(l'_j(\boldsymbol{x}_j) + \delta_j \right) \right\} \\ \text{such that} \quad & \varrho_j \geq 0 \ (j \in \mathcal{J}_i), \quad \sum_{j \in \mathcal{J}_i} \varrho_j = 1. \end{aligned} \tag{2.29}$$

The linear programming dual of (2.29) is $\max_{\alpha \in [0,1]} h_i(\alpha)$ with

$$h_i(\alpha) = \min_{j \in \mathcal{J}_i} \left\{ \alpha(l_j(\boldsymbol{x}_j) - \underline{\Phi}_i) + (1 - \alpha) l'_j(\boldsymbol{x}_j) \ + \delta_j \right\}. \tag{2.30}$$

Algorithm 13 *A partially inexact version of the constrained level method.*

13.0 Parameter setting.

Set the stopping tolerance $\epsilon > 0$.

Set the parameters λ and μ $(0 < \lambda, \mu < 1)$.

Set the accuracy regulating parameter γ such that $0 < \gamma < (1 - \lambda)^2$.

13.1 Bundle initialization.

Let $i = 1$ (iteration counter).

Find a starting point $\boldsymbol{x}_1 \in X$.

Let $l_1(\boldsymbol{x})$ and $l'_1(\boldsymbol{x})$ be linear support functions to $\varphi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$, respectively, at $\boldsymbol{x}_1$.

Let $\delta_1 = 0$ (meaning that l_1 and l'_1 are exact support functions).

Let $\mathcal{J}_1 = \{1\}$ (set of substantial indices).

13.2 Model construction and near-optimality check.

Let $\varphi_i(\boldsymbol{x})$ and $\psi_i(\boldsymbol{x})$ be the current model functions.

Compute the minimum $\underline{\Phi}_i$ of the current model problem (2.6).

Let $h_i(\alpha)$ be the current dual function defined in (2.30).
If $\max\limits_{\alpha\in[0,1]} h_i(\alpha) < \epsilon$, then near-optimal solution found, stop.

13.3 Tuning the dual variable.

Determine the interval $I_i \subseteq [0,1]$ on which h_i takes non-negative values.
Let $\hat{I}_i$ be obtained by shrinking I_i into its center with the factor $(1-\mu)$.

Set α_i according to (2.17).

13.4 Finding a new primal iterate.

Let $\underline{\phi}_i = \alpha_i \underline{\Phi}_i$ and $\overline{\phi}_i = \alpha_i \underline{\Phi}_i + h_i(\alpha_i)$.

Define the level set
$X_i = \left\{ \boldsymbol{x} \in X \,\middle|\, \alpha_i \varphi_i(\boldsymbol{x}) + (1-\alpha_i)\psi_i(\boldsymbol{x}) \leq \underline{\phi}_i + \lambda h_i(\alpha_i) \right\}$.

Let $\boldsymbol{x}_{i+1} = \arg\min\limits_{\boldsymbol{x}\in X_i} \|\boldsymbol{x} - \boldsymbol{x}_i\|^2$.

13.5 Bundle update.

Let $\delta_{i+1} = \gamma h_i(\alpha_i)$.

Call the oracle with the following inputs:
- the current iterate $\boldsymbol{x}_{i+1}$,
- the current dual iterate α_i,
- the accuracy tolerance δ_{i+1}, and
- the descent target $\overline{\phi}_i - \delta_{i+1}$.

Let $l_{i+1}(\boldsymbol{x})$ and $l'_{i+1}(\boldsymbol{x})$ be the linear functions returned by the oracle.

If the descent target was reached then let $\mathcal{J}_{i+1} = \mathcal{J}_i \cup \{i+1\}$,
otherwise let $\mathcal{J}_{i+1} = \mathcal{J}_i$.

Increment i, and repeat from step *13.2*.

Specification 14 *Oracle for Algorithm 13.*

The input parameters are
$\hat{\boldsymbol{x}}$: the current iterate,
$\hat{\alpha}$: the current dual iterate,
$\hat{\delta}$: the tolerance, and
$\hat{\phi}$: the descent target.

The oracle returns linear functions $\ell(\boldsymbol{x})$ and $\ell'(\boldsymbol{x})$ such that

$$\ell(\boldsymbol{x}) \le \varphi(\boldsymbol{x}),\ \ell'(\boldsymbol{x}) \le \psi(\boldsymbol{x}) \quad (\boldsymbol{x} \in X), \quad \|\nabla\ell\|,\ \|\nabla\ell'\| \le \Lambda, \quad \text{and}$$

$$\begin{array}{ll} \text{either} & \hat{\alpha}\ell(\hat{\boldsymbol{x}}) + (1-\hat{\alpha})\ell'(\hat{\boldsymbol{x}}) > \hat{\phi}, \\ & \text{certifying that the descent target cannot be attained,} \\ \text{or} & \hat{\alpha}\ell(\hat{\boldsymbol{x}}) + (1-\hat{\alpha})\ell'(\hat{\boldsymbol{x}}) \le \hat{\phi}, \ \text{in which case} \\ & \ell(\hat{\boldsymbol{x}}) \ge \varphi(\hat{\boldsymbol{x}}) - \hat{\delta} \ \text{ and } \ \ell'(\hat{\boldsymbol{x}}) \ge \psi(\hat{\boldsymbol{x}}) - \hat{\delta} \ \text{ should also hold.} \end{array}$$

The efficiency estimate (2.19) of the constrained level method can be adapted to the partially inexact version:

Theorem 15 *Let $\epsilon > 0$ be a given stopping tolerance. To obtain an ϵ-optimal ϵ-feasible solution of the constrained convex problem (2.2), it suffices to perform $c(\mu,\lambda,\gamma)\left(\frac{2\Lambda D}{\epsilon}\right)^2 \ln\left(\frac{2\Lambda D}{\epsilon}\right)$ iterations, where $c(\mu,\lambda,\gamma)$ is a constant that depends only on the parameters.*

Lemaréchal, Nemirovskii and Nesterov's proof of (2.19) adapts to the partially inexact case. I'm going to show that Propositions 4, 5, 6 apply to the inexact objects defined in this section.

Proof of Proposition 4 adapted to the inexact objects. Let ϱ_j $(j \in \mathcal{J}_i)$ denote an optimal solution of (2.29). Due to linear programming duality, the assumption implies

$$\epsilon > \max_{\alpha\in[0,1]} h_i(\alpha) = \max\left\{ \sum_{j\in\mathcal{J}_i} \varrho_j \left(l_j(\boldsymbol{x}_j) + \delta_j\right) - \underline{\Phi}_i,\ \sum_{j\in\mathcal{J}_i} \varrho_j \left(l'_j(\boldsymbol{x}_j) + \delta_j\right) \right\}.$$

Specifically, we have

$$\epsilon > \sum_{j\in\mathcal{J}_i} \varrho_j \left(l'_j(\boldsymbol{x}_j) + \delta_j\right) \ \ge\ \sum_{j\in\mathcal{J}_i} \varrho_j\, \psi(\boldsymbol{x}_j) \ \ge\ \psi(\boldsymbol{x}_i^\star).$$

The second inequality is a consequence of $l'_j(\boldsymbol{x}_j) + \delta_j \ge \psi(\boldsymbol{x}_j)$ $(j \in \mathcal{J}_i)$. (The third inequality is due to the convexity of the function $\psi(\boldsymbol{x})$, and the construction of $\boldsymbol{x}_i^\star$.)

Near-optimality, i.e., $\epsilon > \varphi(\boldsymbol{x}_i^\star) - \Phi$ can be proven similarly (taking into account $\underline{\Phi}_i \le \Phi$). □

Propositions 5 and 7 are not affected by changing to the inexact objects. (These propositions are based on the concavity of the dual function $h(\alpha)$.)

Instead of Proposition 6, we can use the following analogous form (applying the partially inexact level method instead of the original exact method).

Proposition 16 *Consider a sequence of iterations in the course of which the dual iterate does not change; namely, let $t < s$ be such that $\alpha_t = \cdots = \alpha_s$.*

If $s - t > c(\lambda,\gamma)\left(\frac{\Lambda D}{\varepsilon}\right)^2$ holds with some $\varepsilon > 0$, then $h_s(\alpha_s) \le \varepsilon$ follows. – Here $c(\lambda,\gamma)$ is the constant in the efficiency estimate of Theorem 10.

Proof of Proposition 16. Let $\vartheta(\boldsymbol{x}) = \alpha_s\varphi(\boldsymbol{x}) + (1-\alpha_s)\psi(\boldsymbol{x})$. Then $t_j(\boldsymbol{x}) = \alpha_s l_j(\boldsymbol{x}) + (1-\alpha_s)l'_j(\boldsymbol{x})$ $(j = 1, 2, \ldots)$ satisfy $t_j(\boldsymbol{x}_j) \le \vartheta(\boldsymbol{x})$. Moreover we have $t_j(\boldsymbol{x}_j) + \delta_j \ge \vartheta(\boldsymbol{x})$ for $j \in \mathcal{J}_s$.

Restricting the examination to the iterations $t \le j \le s$, we look on Algorithm 13 as the unconstrained Algorithm 8 applied to the composite objective $\vartheta(\boldsymbol{x})$. It is easy to check that the assignment $\underline{\phi}_j = \alpha_s\underline{\Phi}_j$ and $\overline{\phi}_j = \alpha_s\underline{\Phi}_j + h_j(\alpha_s)$ in step *13.4* satisfies the rules of Remark 12. Indeed, $\overline{\phi}_j \ge \underline{\phi}_j$ due to $h(\alpha_s) \ge 0$, which, in turn, is a consequence of the selection of the dual iterate. Moreover, let $\boldsymbol{u}_j$ denote an optimal solution of the model problem (2.6). Obviously $\vartheta(\boldsymbol{u}_j) = \alpha_s\varphi(\boldsymbol{u}_j) + (1-\alpha_s)\psi(\boldsymbol{u}_j) \le \alpha_s\underline{\Phi}_j = \underline{\phi}_j$. Hence the proposition follows from Theorem 10. □

The partially inexact version of the constrained level method consists of runs of an unconstrained method (namely, a special form of the partially inexact level method.) As we have seen, the convergence proof of the constrained method goes back to Theorem 10.

For the unconstrained methods, I have formulated a special descent target, as a convex combination of the model function value at the new iterate on the one hand, and the best upper estimate known, on the other hand. The weight κ was set in (2.23). This descent target is also inherited to the constrained methods. Applying Proposition 11 to the runs of the unconstrained method, we obtain

Corollary 17 *Let $\kappa = \frac{\gamma}{1-\lambda}$. The efficiency estimate of Theorem 15 remains valid with the descent target $\kappa\Big(\alpha_i\varphi_i(\boldsymbol{x}_{i+1}) + (1-\alpha_i)\psi_i(\boldsymbol{x}_{i+1})\Big) + (1-\kappa)\overline{\phi}_i$ set in step 13.5 of the partially inexact version of the constrained level method.*

The computational study of [64] indicates that the practical efficiency of Algorithm 13 is substantially better than the theoretical estimate of Theorem 15.

2.5 Application of the results

In Chapters 4 and 6, I discuss application of these approximate methods to two-stage stochastic programming problems and to risk-averse variants.

Van Ackooij and de Oliveira in [174] extended the partially inexact version of the constrained level method (of Algorithm 13) to handle upper oracles, i.e., oracles that provide inexact information which might overestimate the exact values of the functions. They cite the research report [53], a former version of [64].

2.6 Summary

In [51], I developed an approximate version of the level method of Lemaréchal, Nemirovskii and Nesterov [99]. The idea was to always set the accuracy tol-

erance in proportion to the current gap. I extended the convergence proof of [99] to the approximate level method. Subsequent computational studies ([62], [184]) indicate that my approximate level method inherits the superior experimental efficiency of the level method. I also worked out an approximate version of the constrained level method of [99], and extended the convergence proof to the approximate version.

My approximate level method was one of the precursors of the 'on-demand accuracy oracle' approach of the Charles Broyden Prize-winning paper of Oliveira and Sagastizábal [27]. The authors of the latter paper implemented a decomposition framework for the solution of two-stage stochastic programming test problems. The master problems were solved with cutting-plane, bundle and level methods, applying different approaches for the handling of inexact data. Regularized methods performed better than pure cutting-plane methods. Among regularized methods, level methods performed best. Inexact function evaluations proved generally effective. The best means of handling inexact data proved to be a combination of my approximate level method and of Kiwiel's partially inexact approach.

In [64] and [184] I worked out a special version of the on-demand accuracy approach of de Oliveira and Sagastizábal [27]. — According to the taxonomy of [27], my method falls into the 'partly asymptotically exact' category, and this term was used also in our papers [64] and [184]. In this dissertation, I call the method 'partially inexact' to keep the terminology simple. (The latter term is in accord with Kiwiel's terminology of [91].)

My method admits a special descent target; a convex combination of the model function value at the new iterate on the one hand, and the best upper estimate known, on the other hand. This setup proved especially effective and interesting in the solution of two-stage stochastic programming problems. The computational study of [184] indicates that the partially inexact level method inherits the superior experimental efficiency of the level method.

In [64], I extended the on-demand accuracy approach to constrained problems. The partially inexact version of the constrained level method consists of runs of an unconstrained method (namely, a special form of the partially inexact level method.) The computational study of [64] indicates that the practical efficiency of the partially inexact version of the constrained level method is substantially better than the theoretical estimate of Theorem 15. We applied this method to the solution of risk-averse two-stage stochastic programming problems.

Van Ackooij and de Oliveira in [174] extended my partially inexact version of the constrained level method to handle upper oracles.

Chapter 3

Cutting-plane methods for risk-averse problems

In this chapter I discuss efficiency issues concerning some well-known means of risk aversion in single-stage models.

3.1 The broader context: comparing and measuring random outcomes

In economics, stochastic dominance was introduced in the 1960's, describing the preferences of rational investors concerning random yields. The concept was inspired by the theory of majorization in Hardy, Littlewood and Pólya [76] who, in turn, refer to Muirhead [110]. Different definitions of what is considered rational result in different dominance relations. Quirk and Saposnik [137] considered first-order stochastic dominance and demonstrated its connection to utility functions. In this dissertation I deal with second-order stochastic dominance that was brought to economics by Hadar and Russel [74]. – Recent applications of second-order stochastic dominance-based models are discussed in [46], [179].

Let $\mathcal{R}$ denote the space of legitimate random losses. A risk measure is a mapping $\rho : \mathcal{R} \to [-\infty, +\infty]$. The acceptance set of a risk measure ρ is defined as $\{R \in \mathcal{R} \,|\, \rho(R) \leq 0\}$. Artzner et al. [6] argued that reasonable risk measures have convex cones as acceptance sets. They characterized these risk measures and introduced the term coherent for them. A classic example of a coherent risk measure is the conditional value-at-risk that I'm going to discuss in more detail.

3.2 Conditional value-at-risk and second-order stochastic dominance

Let R denote a random variable representing uncertain yield or loss. We assume that the expectation of R exists. In a decision model, the random yield or loss is a function of a decision vector $\boldsymbol{x} \in \mathbb{R}^n$. We use the notation $R(\boldsymbol{x})$. The feasible domain will be denoted by $X \subset \mathbb{R}^n$ that we assume to be a convex polyhedron.

We focus on discrete finite distributions, where realizations of $R(\boldsymbol{x})$ will be denoted by $r_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$, and the corresponding probabilities by p_s $(s = 1, \ldots, S)$. We assume that the functions $r_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$ are linear.

Expected shortfall. R represents uncertain yield in this case. Given $t \in \mathbb{R}$ let us consider $\mathrm{E}\left([t - R]_+\right)$, where $[.]_+$ denotes the positive part of a real number. This expression can be interpreted as expected shortfall with respect to the target t, and will be denoted by $\mathrm{ES}_t(R)$. (Though the term 'expected shortfall' is also used in a different meaning, especially in finance.)

In a decision model, we can add a constraint in the form $\mathrm{ES}_t(R(\boldsymbol{x})) \leq \rho$ with a constant $\rho \in \mathbb{R}_+$. Constraints of this type were introduced by Klein Haneveld [92], under the name of integrated chance constraints.

In case of discrete finite distributions, an obvious way of constructing a linear representation of the integrated chance constraint is by introducing a new variable to represent $[t - r_s(\boldsymbol{x})]_+$ for each $s = 1, \ldots, S$. We will call this lifting representation.

Klein Haneveld and Van der Vlerk [93] proposed the following polyhedral representation

$$\sum_{s=1}^{S} p_s [t - r_s(\boldsymbol{x})]_+ = \max_{J \subset \{1,\ldots,S\}} \sum_{s \in J} p_s \left(t - r_s(\boldsymbol{x})\right) \qquad (\boldsymbol{x} \in \mathbb{R}^n). \tag{3.1}$$

Based on the above representation, Klein Haneveld and Van der Vlerk implemented a cutting-plane method for the solution of integrated chance constrained problems. They compared this approach with the lifting representation, where the resulting problems were solved with a benchmark interior-point solver. On smaller problem instances, the cutting-plane algorithm could not beat the interior-point solver. However, the cutting-plane approach proved much faster on larger instances.

Tail expectation and Conditional Value-at-Risk (CVaR). Given a random yield R and a probability β $(0 < \beta \leq 1)$, let $\mathrm{Tail}_\beta(R)$ denote the unconditional expectation of the lower β-tail of R. – This is the same as the second quantile function introduced by Ogryczak and Ruszczyński in [118].

Now let R represent uncertain loss or cost. Given a confidence level $(1 - \beta)$ such that $0 < \beta \leq 1$, the risk measure $\mathrm{CVaR}_\beta(R)$ is the conditional expectation of the upper β-tail of R. Obviously we have

$$\beta\, \mathrm{CVaR}_\beta(R) = -\mathrm{Tail}_\beta(-R), \tag{3.2}$$

where $-R$ now represents random yield.

The CVaR risk measure was characterized by Rockafellar and Uryasev [143, 144], and Pflug [123]. The former authors in [143] established the minimization rule

$$\mathrm{CVaR}_\beta(R) = \min_{t\in\mathbb{R}} \left\{ t + \frac{1}{\beta}\mathrm{E}\left([R-t]_+\right) \right\} \tag{3.3}$$

that is widely used in CVaR-minimization models.

Considering R a random yield, Ogryczak and Ruszczyński [118] established the convex conjugacy relation

$$\mathrm{Tail}_\beta(R) \;=\; \max_{t\in\mathbb{R}} \left\{ \beta t - \mathrm{ES}_t(R) \right\} \tag{3.4}$$

which, in view of (3.2), is obviously equivalent to (3.3). The latter authors also present CVaR minimization as a two-stage stochastic programming problem.

The CVaR risk measure originally comes from finance (where it is now widely used), and is getting applied in other areas, see, e.g., [116].

In a decision model of discrete finite distribution, the lifting representation is an obvious way of formulating CVaR computation as a linear programming problem. It means introducing in (3.3) a new variable to represent $[r_s(\boldsymbol{x}) - t]_+$ for each $s = 1, \ldots, S$.

An alternative, polyhedral, representation was proposed by Künzi-Bay and Mayer [97] who showed that

$$\begin{aligned} \mathrm{CVaR}_\beta\big(R(\boldsymbol{x})\big) = \quad & \min\ t + \tfrac{1}{\beta}\vartheta \\ & \text{such that} \quad t, \vartheta \in \mathbb{R}, \quad \text{and} \\ & \sum_{s\in J} p_s \left(r_s(\boldsymbol{x}) - t\right) \le \vartheta \quad \text{for each } J \subset \{1, \ldots, S\} \end{aligned} \tag{3.5}$$

holds for any $\boldsymbol{x}$. Of course this is an analogue of (3.1), but Künzi-Bay and Mayer obtained it independently, through formulating CVaR minimization as a two-stage stochastic programming problem.

Based on the above representation, Künzi-Bay and Mayer implemented a cutting-plane method for the solution of CVaR-minimization problems. They compared this approach with the lifting representation, where the resulting problems were solved with general-purpose LP solvers. Problems were solved with increasing numbers of scenarios, and the results show that the cutting-plane approach has superior scale-up properties. For the larger test problems, it was by 1-2 orders of magnitude faster than the lifting approach.

Remark 18 *Given random loss R, the measure $CVaR_\beta(R)$ is often defined as the conditional expectation of the upper $(1-\beta)$-tail instead of the β-tail, especially if the intention is to compare CVaR and VaR. Moreover, CVaR is often defined for a random yield, instead of random loss.*

Differing definitions were used also in our works [52], [57], [58], [59].

Second-order stochastic dominance and a dominance measure. Let R and R' represent uncertain yields. We assume that the expectation of R' also exists. We say that R dominates R' with respect to second-order stochastic dominance, and use the notation $R \succeq_{SSD} R'$, if either of the following equivalent conditions hold:

(a) $\mathrm{E}\,(u(R)) \geq \mathrm{E}\,(u(R'))$ holds for any nondecreasing and concave utility function u for which these expected values exist and are finite.

(b) $\mathrm{ES}_t(R) \leq \mathrm{ES}_t(R')$ holds for each $t \in \mathbb{R}$.

(c) $\mathrm{Tail}_\beta(R) \geq \mathrm{Tail}_\beta(R')$ holds for each $0 < \beta \leq 1$.

Concavity of the utility function in *(a)* characterizes risk-averse behavior. The equivalence of *(a)* and *(b)* has been known long ago; see e.g. [181]. The equivalence of *(b)* and *(c)* has been shown by Ogryczak and Ruszczyński [118] as a consequence of (3.4). In general, SSD relations can be described with a continuum of constraints.

Let us assume that a reference return $\widehat{R}$ is available (an integrable random variable of known distribution). Dentcheva and Ruszczyński in [41] and [42] introduced SSD constraints $R(\boldsymbol{x}) \succeq_{SSD} \widehat{R}$ in stochastic models and explored mathematical properties of the resulting optimization problems for general distributions. These authors also develop a duality theory in which dual objects are nondecreasing concave utility functions. They prove that, in case $\widehat{R}$ has discrete finite distribution, the SSD relation can be characterized by a finite system of inequalities of type *(b)*.

Roman, Darby-Dowman, and Mitra in [147] use criterion *(c)*. They assume finite discrete distributions with equally probable outcomes, and prove that, in this case, the SSD relation can be characterized by a finite system of inequalities. Namely, prescribing the tail inequalities for $\beta = \frac{s}{S}$ $(s = 1, \ldots, S)$ is sufficient. Based on this observation, they propose choosing $\boldsymbol{x} \in X$ such that the return $R(\boldsymbol{x})$ comes close to, or emulates, the reference return $\widehat{R}$ in a uniform sense. Uniformity is meant in terms of differences among tails; i.e., the 'worst' tail difference

$$\min_{1 \leq s \leq S} \left\{ \mathrm{Tail}_{\frac{s}{S}}\big(R(\boldsymbol{x})\big) - \mathrm{Tail}_{\frac{s}{S}}\big(\widehat{R}\big) \right\} \tag{3.6}$$

is maximized over X. This can be considered a multi-objective model whose origin can be traced back to [117].

3.3 Contribution

The convex conjugacy relationship (3.4) reduces to linear programming duality in case of discrete finite distributions, as I worked out in [52]. Given $\boldsymbol{x} \in \mathbb{R}^n$, the tail expectation of the corresponding yield can be computed as the optimum

of the linear programming problem

$$\begin{aligned} \mathrm{Tail}_\beta\left(R(\boldsymbol{x})\right) = \quad & \min \sum_{s=1}^{S} \pi_s\, r_s(\boldsymbol{x}) \\ & \text{such that} \quad 0 \le \pi_s \le p_s \;\; (s = 1, \ldots, S), \quad \text{and} \\ & \sum_{s=1}^{S} \pi_s = \beta, \end{aligned} \tag{3.7}$$

where the decision variable π_s means the weight of the sth scenario in the lower β-tail. The linear programming dual of (3.7) can be transformed into

$$\max_{t \in \mathbb{R}} \left\{ \beta t - \sum_{s=1}^{S} p_s \left[t - r_s(\boldsymbol{x})\right]_+ \right\} \tag{3.8}$$

which is just the convex conjugate of the expected shortfall (the latter considered as a function of the target t).

Using (3.2), problem (3.7) can be formulated with CVaR instead of Tail:

$$\begin{aligned} \mathrm{CVaR}_\beta\left(R(\boldsymbol{x})\right) = \quad & \max \sum_{s=1}^{S} \varpi_s\, r_s(\boldsymbol{x}) \\ & \text{such that} \quad 0 \le \varpi_s \le \tfrac{p_s}{\beta} \;\; (s = 1, \ldots, S), \quad \text{and} \\ & \sum_{s=1}^{S} \varpi_s = 1. \end{aligned} \tag{3.9}$$

(3.9) turned out to be a discrete version of the risk envelope of [145]. The above discrete formulation proved effective for handling CVaR constraints in two-stage problems, as I'm going to report in Chapter 6. (A dual solution approach, also based on the above formulation, was proposed in [63].)

In the special case of $p_s = \frac{1}{S}$ $(s = 1, \ldots, S)$ and $\beta \in \left\{\frac{1}{S}, \frac{2}{S}, \ldots, 1\right\}$, basic solutions of (3.9) have components $\pi_s = 0$ or $\frac{1}{S}$ $(s = 1, \ldots, S)$. Hence (3.9) reduces to

$$\begin{aligned} \mathrm{CVaR}_\beta(R(\boldsymbol{x})) = \quad & \max_J \; \tfrac{1}{\beta S} \sum_{s \in J} r_s(\boldsymbol{x}) \\ & \text{such that} \quad J \subset \{1, \ldots, S\}, \;\; |J| = \beta S. \end{aligned} \tag{3.10}$$

This formula can be considered an adaptation of the polyhedral representation of Künzi-Bay and Mayer. The variable t of (3.5) becomes superfluous in the equiprobable case, and cuts belonging to sets of cardinality βS are sufficient.

I worked out cutting-plane approaches for the handling of SSD in stochastic programming problems. These were implemented and investigated in collaboration with Gautam Mitra and Diana Roman of CARISMA (Centre for the

Analysis of Risk and Optimisation Modelling Applications) from Brunel University, London. We implemented a solution method for the uniform-dominance model (3.6) of Roman, Darby-Dowman, and Mitra. The method was based on the polyhedral representation (3.10). Algorithmic descriptions and test results were presented in [57]. The cutting-plane approach resulted in dramatic improvement in efficiency; portfolio-optimization problems were solved in seconds instead of hours. My co-authors formerly used a solver based on lifting representations which took several hours to solve problems with $n = 76$ and $S = 500$. Solution time sharply increased with further increase in the number of the scenarios. The cutting-plane based solver solved these problems in a few seconds, and showed good scale-up behaviour: even with $S = 10.000$ scenarios, it solved the problems within ten seconds.

Rudolf and Ruszczyński in [149] also developed cutting-plane approaches for the handling of SSD constraints. The stochastic programing community accepts that our results are independent. (An early version of [57] was published in the same year as [149].)

I proposed a scaled version of the uniform-dominance model (3.6). A new decision variable $\vartheta \in \mathbb{R}$ was introduced, representing a 'certain' (i.e., riskless) yield. (In a portfolio optimization example, this means holding an amount of cash.) Consider the dominance measure

$$\max \left\{ \vartheta \in \mathbb{R} \;\middle|\; R(\boldsymbol{x}) \succeq_{SSD} \widehat{R} + \vartheta \right\}. \tag{3.11}$$

In a portfolio-optimization example, the above SSD-relation means that we prefer the return $R(\boldsymbol{x})$ to the combined return of the stock index and ϑ amount of cash. – The construction is analogous to that of certain risk measures, and the negative of this dominance measure turns out to be a convex risk measure in the sense of Rockafellar [142].

In a portfolio-optimization problem, the measure (3.11), as a function of $\boldsymbol{x}$, is maximized such that $\boldsymbol{x} \in X$.

In view of definition *(c)* of the second-order stochastic dominance, the relation $R(\boldsymbol{x}) \succeq_{SSD} \widehat{R} + \vartheta$ is equivalent to

$$\mathrm{Tail}_\beta(R(\boldsymbol{x})) \;\geq\; \mathrm{Tail}_\beta(\widehat{R}) \;+ \beta\vartheta \tag{3.12}$$

holding for $0 < \beta \leq 1$. Using (3.2), the above inequality naturally transforms to CVaR. In the equiprobable case the cutting-plane representation (3.10) can be applied.

We compared modeling aspects of the dominance measures (3.6) and (3.11) in collaboration with Gautam Mitra, Diana Roman and Victor Zverovich from Brunel University. Algorithmic descriptions and test results were presented in [58]. This study confirmed a shape-preserving quality of the dominance measure (3.11). The resulting optimal portfolio $\boldsymbol{x}^\star$ has a yield $R(\boldsymbol{x}^\star)$, the shape of whose distribution is similar to that of the reference return.

A more thorough computational study was presented in the book chapter [59]. My co-workers were Gautam Mitra, Diana Roman and Victor Zverovich

from Brunel University; and Tibor Vajnai, Edit Csizmás and Olga Papp from Kecskemét College. Our input data set consisted of weekly returns of 68 stocks from the FTSE 100 basket, together with the FTSE 100 index returns. We partitioned the observed weeks into subsets $\mathcal{H}$ and $\mathcal{T}$. The subset $\mathcal{H}$ was used for portfolio construction. Returns corresponding to $\mathcal{H}$ were considered as equally probable scenarios. We maximised the unscaled dominance measure (3.6) and the scaled dominance measure (3.11), respectively, over the simplex $X = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} \geq \mathbf{0},\ \sum x_i = 1\}$. The index played the role of reference return $\widehat{R}$. We then used the subset $\mathcal{T}$ for out-of-sample tests. Considering the returns corresponding to $\mathcal{T}$ as equally probable scenarios, we constructed return histograms of the respective optimal portfolios of the unscaled end the scaled model.

We repeated the above experiment 12 times, always partitioning our dataset into subsets $\mathcal{H}$ and $\mathcal{T}$ in a random manner. The following observations hold in each individual experiment we performed: The index histogram has a longish left tail. The unscaled histogram is curtailed on the left. The tail of the scaled histogram is similar to that of the index. The unscaled histogram has significantly larger expectation than the index histogram. The scaled histogram, in turn, has significantly larger expectation than the unscaled one, though its standard deviation is somewhat larger also. These observations point to the applicability of the scaled model.

3.4 Application of the results

The models and solvers developed in the course of the above mentioned projects have been included in the optimization and risk analytics tools developed at OptiRisk Systems, http://www.optirisk-systems.com. OptiRisk is an informatics and consulting company specializing in risk management, and utilizing the results of research done at Brunel University.

My former co-workers Roman, Mitra and Zverovich in [148] performed a systematic comparison of the unscaled model (3.6) and the scaled model (3.11). The following paragraphs are cited from this paper.

> We have tested the effectiveness of these two models as enhanced indexation strategies, using three datasets: FTSE 100 (97 stocks), Nikkei 225 (222 stocks) and SP 500 (491 stocks). We have used the last half of 2011 (01/06/11–22/12/11) as a backtesting period, in a daily rebalancing frame: for each model and each market we have computed 147 ex-post compounded returns. These are "realised" returns: portfolio strategies are implemented and then evaluated on the next time period using real data. We have made a comparison with the indices' performance and also with the performance of the index tracker portfolios obtained with Roll's (1992) model [146].
>
> Three conclusions are drawn.
>
> First, the SSD-based models consistently outperform the corresponding indices, in the sense that higher returns are obtained over most of

> the backtesting period. This aspect is emphasised by computing their compounded returns. All the three indices are generally at loss over the backtesting period, with the index trackers mimicking nearly perfectly their movements. In contrast, the portfolios obtained with the SSD models lead to overall profits. In particular, portfolios obtained via the SSD scaled model have a very good backtesting performance, consistently outperforming the corresponding indices (also the SSD unscaled portfolios) by a substantial amount. For all three markets, the SSD scaled strategy results in a compounded gain of 40% or above, while the indices have a compounded loss around 10%. ...
>
> Secondly, the imposition of cardinality constraints seems to be unnecessary in the two SSD-based models. Due to their CVaR-minimisation nature, these models naturally select a much lower number of stocks than the established index tracking models. ...
>
> Finally, the amount of necessary rebalancing in the SSD-based models is low, since the models are stable with the introduction of new scenarios, representing new information on the market. ...

Novel approaches for portfolio construction were proposed by Valle, Mitra and Roman in [172]. This is a sequel to [148] and the enhancements are based on the scaled model.

Enhanced versions of the cutting-plane method described in [57] were developed by Sun et al. [160] and Khemchandani et al. [88].

3.5 Summary

The convex conjugacy relationship between expected shortfall and tail expectation reduces to linear programming duality in case of discrete finite distributions, as I worked out in [52]. This approach yields a CVaR formulation that proved effective for handling CVaR constraints in two-stage problems (reported in [64]).

I worked out cutting-plane approaches for the handling of SSD in stochastic programming problems. These were implemented and investigated in collaboration with Gautam Mitra and Diana Roman from Brunel University. Algorithmic descriptions and test results were presented in [57]. The cutting-plane approach resulted in dramatic improvement in efficiency; portfolio-optimization problems were solved in seconds instead of hours.

I proposed a scaled version of the uniform-dominance model of Roman, Darby-Dowman, and Mitra. In a portfolio optimization example, the scaled dominance relation means that we prefer the return of our portfolio to the combined return of a benchmark portfolio and a certain amount of cash.

We compared modeling aspects of the scaled and the unscaled dominance measures in collaboration with Gautam Mitra, Diana Roman and Victor Zverovich from Brunel University. Algorithmic descriptions and test results were presented in [58]. This study confirmed a shape-preserving quality of the scaled dominance

measure: the resulting optimal portfolio has a yield the shape of whose distribution is similar to that of the reference return.

The models and solvers developed in the course of the above mentioned projects have been included in the optimization and risk analytics tools developed at OptiRisk Systems, an informatics and consulting company specializing in risk management, and utilizing the results of research done at Brunel University.

My former co-workers Roman, Mitra and Zverovich in [148] performed a systematic comparison of the unscaled model and the scaled one. They observe that 'portfolios obtained via the SSD scaled model have a very good backtesting performance, consistently outperforming the corresponding indices (also the SSD unscaled portfolios) by a substantial amount'.

Novel approaches for portfolio construction were proposed by Valle, Mitra and Roman in [172]. This is a sequel to [148] and the enhancements are based on the scaled model.

Enhanced versions of the cutting-plane method described in [57] were developed by Sun et al. [160] and Khemchandani et al. [88].

Chapter 4

Decomposition methods for two-stage stochastic programming problems

Two-stage stochastic programming problems derive from such models where decisions are made in two stages and the observation of some random event takes place in between. Hence the first decision must be made when the outcome of the random event is not yet known. For example, the first stage may represent the decision on the design of a system; and the second stage, a decision on the operation of the system under certain circumstances. The aim is to find a balance between investment cost and long-term operation costs.

In this chapter I confine discussion to the case when no feasibility issues occur (as formulated in Assumption 19, below.) Feasibility issues will be considered in Chapter 5.

4.1 The classic two-stage SP problem

The model originates from Dantzig [24] and Beale [7], and mathematical characterization was given by Wets [180].

The first-stage decision is represented by the vector $\boldsymbol{x} \in X$, the feasible domain being defined by a set of linear inequalities. We assume that the feasible domain is a nonempty convex bounded polyhedron, and that there are S possible outcomes (scenarios) of the random event, the sth outcome occurring with probability p_s.

Suppose the first-stage decision has been made with the result $\boldsymbol{x}$, and the sth scenario has realized. The second-stage decision $\boldsymbol{y}$ is computed by solving the second-stage problem or recourse problem that we denote by $\mathcal{R}_s(\boldsymbol{x})$. This

is a linear programming problem whose dual is $\mathcal{D}_s(\boldsymbol{x})$:

$$\begin{array}{l|l} \mathcal{R}_s(\boldsymbol{x}) & \begin{array}{l}\min \boldsymbol{q}_s^T \boldsymbol{y} \\ \text{such that} \\ T_s\boldsymbol{x} + W_s\boldsymbol{y} = \boldsymbol{h}_s, \\ \boldsymbol{y} \geq \boldsymbol{0},\end{array} \\ \hline \mathcal{D}_s(\boldsymbol{x}) & \begin{array}{l}\max \boldsymbol{z}^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) \\ \text{such that} \\ W_s^T\boldsymbol{z} \leq \boldsymbol{q_s}, \\ \boldsymbol{z} \text{ is a real-valued vector.}\end{array} \end{array} \tag{4.1}$$

In the above formulae, $\boldsymbol{q}_s$, $\boldsymbol{h}_s$ are given vectors and T_s, W_s are given matrices, with compatible dimensions. In this chapter we work under

Assumption 19 (relatively complete recourse) *The recorse problem $\mathcal{R}_s(\boldsymbol{x})$ is feasible for any $\boldsymbol{x} \in X$ and $s = 1, \ldots, S$.*

Moreover we assume that $\mathcal{D}_s(\boldsymbol{x})$ is feasible for any $s = 1, \ldots, S$. Let $q_s(\boldsymbol{x})$ denote the common optimum. This is a polyhedral convex function called the *recourse function.*

The customary formulation of the *first-stage problem* is

$$\min \ \boldsymbol{c^T x} + \sum_{s=1}^{S} p_s \, q_s(\boldsymbol{x}) \quad \text{such that } \boldsymbol{x} \in X. \tag{4.2}$$

The expectation part of the objective, $q(\boldsymbol{x}) = \sum_{s=1}^{S} p_s \, q_s(\boldsymbol{x})$, is called the *expected recourse function.*

Let us assume that the feasible domain X is defined by a finite set of linear equations, in the form of $A\boldsymbol{x} = \boldsymbol{b}$. The two-stage stochastic programming problem (4.2)-(4.1) can be formulated as a single linear programming problem called the equivalent LP problem:

$$\begin{array}{rccccccccc} \min & \boldsymbol{c}^T\boldsymbol{x} & + & p_1\boldsymbol{q}_1^T\boldsymbol{y}_1 & + & \ldots & + & p_S\boldsymbol{q}_S^T\boldsymbol{y}_S & & \\ \text{subject to} & A\boldsymbol{x} & & & & & & & = & \boldsymbol{b}, \\ & T_1\boldsymbol{x} & + & W_1\boldsymbol{y}_1 & & & & & = & \boldsymbol{h}_1, \\ & \vdots & & & & \ddots & & & & \vdots \\ & T_S\boldsymbol{x} & + & & & & & W_S\boldsymbol{y}_S & = & \boldsymbol{h}_S, \end{array} \tag{4.3}$$

$$\boldsymbol{x} \geq \boldsymbol{0}, \qquad \boldsymbol{y}_s \geq \boldsymbol{0} \ \ (s = 1, \ldots, S).$$

This linear programming problem has a specific structure: for each scenario, a subproblem is included that describes the second-stage decision associated with the corresponding scenario realization. The subproblems are linked by the first-stage decision variables.

Polyhedral models. Given a finite subset $\widetilde{U}_s$ of the feasible domain of $\mathcal{D}_s(\boldsymbol{x})$, the function

$$\widetilde{q}_s(\boldsymbol{x}) = \max_{\boldsymbol{u_s} \in \widetilde{U}_s} \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \quad (\boldsymbol{x} \in X) \tag{4.4}$$

is a lower approximation of $q_s(\boldsymbol{x})$ over X. Having appropriate subsets $\widetilde{U}_s$ for $s = 1, \ldots, S$, the *disaggregate model* of the first-stage problem (4.2) is constructed as

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \sum_{s=1}^{S} p_s \nu_s \\ \text{such that} & \boldsymbol{x} \in X, \quad \nu_s \in \mathbb{R} \ \ (s = 1, \ldots, S), \\ & \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s \boldsymbol{x}) \le \nu_s \ \text{ holds for any } \boldsymbol{u_s} \in \widetilde{U}_s \quad (s = 1, \ldots, S). \end{array} \tag{4.5}$$

The expectation in the objective, $\widetilde{q}(\boldsymbol{x}) = \sum_{s=1}^{S} p_s \, \widetilde{q}_s(\boldsymbol{x})$, is called the disaggregate model function.

An *aggregate model* of the first-stage problem is

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \nu \\ \text{such that} & \boldsymbol{x} \in X, \quad \nu \in \mathbb{R}, \\ & \sum_{s=1}^{S} p_s \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s \boldsymbol{x}) \le \nu \ \text{ holds for any } (\boldsymbol{u_1}, \ldots, \boldsymbol{u_S}) \in \widetilde{\mathcal{U}}, \end{array} \tag{4.6}$$

where $\widetilde{\mathcal{U}} \subset \widetilde{U}_1 \times \cdots \times \widetilde{U}_S$ is a certain subset of the Cartesian product. Namely, each element of $\widetilde{\mathcal{U}}$ belongs to a (potential) facet in the graph of the function $\widetilde{q}(\boldsymbol{x})$. There may be facets not represented in $\widetilde{\mathcal{U}}$. The upper cover

$$\widetilde{f}(\boldsymbol{x}) = \max_{(\boldsymbol{u_1}, \ldots, \boldsymbol{u_S}) \in \widetilde{\mathcal{U}}} \sum_{s=1}^{S} p_s \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s \boldsymbol{x}) \tag{4.7}$$

is called the aggregate model function.

Of course we have $q(\boldsymbol{x}) \ge \widetilde{q}(\boldsymbol{x}) \ge \widetilde{f}(\boldsymbol{x})$, since the disaggregate model function is based on the sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$, and the aggregate model function is based on the set $\widetilde{\mathcal{U}} \subset \widetilde{U}_1 \times \cdots \times \widetilde{U}_S$.

4.2 Solution approaches

A straightforward approach is the application of a general-purpose LP solver to the equivalent LP problem (4.3).

Dantzig and Madansky [25] observed that the dual of the equivalent LP problem fits the prototype for the Dantzig-Wolfe decomposition [26]. The resulting decomposition approach is equivalent to a cutting-plane method based on the disaggregate model (4.5).

The L-shaped method of Van Slyke and Wets [176] is a cutting-plane method based on the aggregate model (4.6). This approach turned out to be identical to a Benders decomposition [8], specially adapted to the equivalent LP problem. (Aggregation being the speciality of the adaptation.)

The Dantzig-Wolfe decomposition was proposed in 1960 to overcome the bottleneck of restricted computer memory. Memory is no longer a scarce resource, but decomposition is still an effective approach, as I'm going to demonstrate.

The distinction between the Dantzig-Wolfe decomposition and the Benders decomposition or the cutting-plane approach is just a question of viewpoint. The advantage of the cutting-plane viewpoint is that it gives a clear visual impression of the procedure and, more importantly, that it enables the application of the enhancements discussed in Chapter 2.

An overview of decomposition methods. The regularized decomposition method of Ruszczyński [150] is a bundle-type method for the minimization of the sum of polyhedral convex functions over a convex polyhedron. Hence it is naturally applied to the disaggregate model. (Ruszczyński developed a bundle reduction mechanism, and keeps only two active cuts per scenario in the master problem.)

The box-constrained trust-region method of Linderoth and Wright [102] also solves the disaggregate problem, using a special trust-region approach.

Zakeri et al [187] applied their simple inexact cut method (mentioned in Chapter 2) in a plain cutting-plane approach to the aggregate problem.

In [62], we applied the approximate level method of [51] (recounted in Chapter 2) to the aggregate problem. I'm going to describe this approach in more detail in Section 4.3.

Oliveira, Sagastizábal and Scheimberg [119] proposed a special inexact bundle method for the solution of the aggregate problem. Comparing their method to our [62], they point out two differences: first, instead of scenario approximation, they apply sampling; second, their method does not explicitly control the oracle inaccuracy.

The difference between disaggregate and aggregate formulations is but technical; yet it results in a substantial difference in efficiency. By using a disaggregate formulation, more detailed information is stored in the master problem. This is done at the expense of larger master problems. Based on the numerical results of [9] and [69], Birge and Louveaux [10] conclude that the disaggregate formulation is in general more effective when the number of the scenarios is not significantly larger than the number of the constraints in the first-stage problem. We refined this observation in the computational study [190] that I'm going to discuss below. – In this computational study we also showed that the equivalent LP formulations of many industrial problems would resist the direct application of even the most powerful general-purpose LP solvers.

In order to find a balance between the size of the master problem and the amount of information stored in it, intermediate approaches between disaggregate and aggregate models have been proposed. Trukhanov et al. [170] proposed an adaptive aggregation method. The idea was to start with a low level of cut aggregation and increasing it over the course of the solution process. Wolf and Koberstein [185] provided further insights into the effects of cut aggregation. Moreover they introduced a technique called cut consolidation that means discarding inactive cuts and adding their aggregation to the master problem.

De Oliveira and Sagastizábal [27] proposed working with an aggregate master problem, while storing in the oracle all the (disaggregate) information obtained

from the solution of the recourse problems. For the solution of the master problem they applied their on-demand accuracy approach recounted in Chapter 2. At an oracle call, no recourse problems are solved if the information stored in the oracle is sufficient to construct a cut of prescribed accuracy.

More recently Song and Luedtke in [157] developed an approximation scheme that is between the disaggregate and the aggregate approaches. Like our [62], it is based on an adaptive partition of the scenarios. Song and Luedtke found their approach to be often competitive with, and occasionally superior to, our method.

On randomized methods. Though I focus on deterministic solution methods in this chapter, allow me a glance at the broader context.

Ermoliev and Shor [48] applied a stochastic quasigradient method to the solution of two-stage stochastic programming problems.

A randomized decomposition scheme was developed by Higle and Sen [80], a detailed overview can be found in their book [81].

Shapiro and Homem-de-Mello in [156] applied a general simulation framework to two-stage stochastic programming problems, and the empirical behavior of the approach was discussed by Linderoth, Shapiro and Wright in [101]. A detailed overview of the approach can be found in [155]. – The sampled problems in this simulation framework can be solved by deterministic methods.

Deák in [32], [33] applied his successive regression approximation to two-stage stochastic programming problems, and discussed the empirical behavior of the approach. An improvement on the approach is proposed by Deák, Pólik, Prékopa and Terlaky in [35].

4.3 Contribution

We completed three projects with different co-workers. In the first project, I adapted the approximate level method to the solution of the aggregate master problem in a decomposition framework. The resulting procedure was named level decomposition. We implemented it with Zoltán Szőke who, at that time, was my doctoral student at the Doctoral School of Mathematics, Eötvös Loránd University.

The aim of the second project was extensive experimentation and solver development at the Brunel University, London. My co-workers were Gautam Mitra, Francis Ellison and Victor Zverovich of the CARISMA team (Centre for the Analysis of Risk and Optimisation Modelling Applications).

In the third project I developed a decomposition framework that combines the advantages of the aggregate and the disaggregate model. We implemented the method and performed an extensive computational study in collaboration with Leena Suhl, Achim Koberstein and Christian Wolf from the DS&OR (Decision Support & Operations Research) Lab of Paderborn University. The aim of our collaboration with the Paderborn team had been the development of effective solvers for a real-life gas purchase problem.

Level decomposition

I adapted the approximate level method [51] (recounted in Chapter 2), to the aggregate master problem. The (inexact) oracle was based on a successive approximation of the distribution.

We implemented the method with my doctoral student Zoltán Szőke at the Eötvös Loránd University. The method was described and a computational study was presented in [62]. Numerical results show that this approximation framework is effective: although the number of the master iterations is larger than in the case of the exact method, there is a substantial reduction in the solution time of the second-stage problems.

To approximate the recourse function, we used the classic distribution approximation framework (described, e.g., in [11]). Fixed recourse was assumed, i.e., $W_s = W$ $(s = 1, \dots, S)$. The space of the right-hand sides of the recourse problems was partitioned into multidimensional intervals. Given a first-stage solution $\boldsymbol{x}$, the right-hand sides $\boldsymbol{h}_s - T_s\boldsymbol{x}$ $(s = 1, \dots, S)$ that fell into the same interval where accumulated into their barycenter.

The structure of our level decomposition implementation is shown in Figure 4.1. The novelty of the approach is that the accuracy of the distribution approximation is regulated by the optimization method applied to the master problem. This setup helps in finding a balance between different efforts; and is in contrast to the classic distribution approximation framework where the main module is the distribution approximation scheme, and the submodule is a two-stage stochastic problem solver. To what extent the approximation should be refined between solver calls, is a difficult question in the classic setup. – Mayer [108] proposed a heuristic method for the regulation of the accuracy of the approximation in that context.

A technical issue had to be solved to make the approximate level method applicable in the decomposition framework: the oracle must return linear functions whose slope remains under a common bound, i.e., $\|\nabla\ell\| \leq \Lambda$ must hold with fixed Λ for every linear function $\ell(\boldsymbol{x})$ returned. I ensured this by always selecting basic solutions of the appropriate dual recourse problems into the sets $\widetilde{U}_s$ $(s = 1, \dots, S)$. Existence of a common bound follows from the fact that the number of the basic solutions is finite. – Note that the distribution approximation scheme does not interfere with this technique; the right-hand side of a recourse problem is the objective vector in the corresponding dual recourse problem. Hence the set of the basic solutions of the dual recourse problem is not affected by the distribution approximation.

Of course the bound Λ cannot be computed in general. But this bound is only needed in the convergence proof, and the level method can be implemented without knowing it. In the computational study [62] we found the method effective, and also observed that the approximate level method inherits the practical efficiency estimate (2.13). Whenever we solved a problem with higher and higher accuracy, a constant number of additional iterations always yielded a further accurate digit in the optimum.

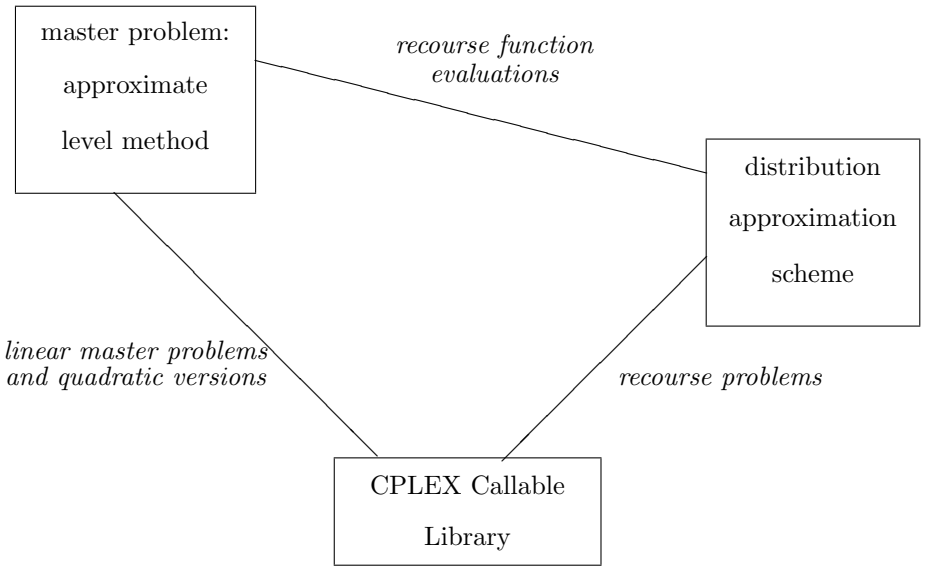

Figure 4.1: *The structure of our level decomposition implementation.*

Experimentation and solver development at Brunel University

The motivation of the computational study [190] was to re-assess different solution approaches to two-stage problems, in view of recent advances in computer architecture and LP solver algorithms. (There was a certain amount of complacency in the stochastic programming community that realistic instances of two-stage problems can solved directly in equivalent LP forms.)

My co-workers were Gautam Mitra, Francis Ellison and Victor Zverovich of the CARISMA team (Centre for the Analysis of Risk and Optimisation Modelling Applications) from Brunel University, London. This was one of the leading teams in computational stochastic programming world-wide. In our experiments we used the software systems developed at CARISMA and OptiRisk, and also extended and enhanced these systems in the course of the project.

The following methods were compared:

Simplex: direct solution of the equivalent LP problem with the CPLEX dual simplex solver.

IPM: direct solution of the equivalent LP problem with an interior-point method. We generally used the CPLEX barrier solver, but also experimented with the HOPDM solver [72], [23]. The latter is an infeasible primal-dual interior-point solver, developed by J. Gondzio and team at the University of Edinburgh.

Benders: plain decomposition based on the aggregate model.

RD: the regularized decomposition method of Ruszczyński [150], implemented with the enhancements of [152].

TR: the box-constrained trust-region method of Linderoth and Wright [102].

Level: decomposition based on the aggregate model, having the master problem solved with an adaptation of the level method of Lemaréchal, Nemirovskii and Nesterov [99]. We have set $\lambda = 0.5$.

In the decomposition frameworks, the CPLEX dual simplex and quadratic solvers were used.

Test problems were drawn from the following sources: the POSTS collection of Holmes [82]; the Slptestset collection of Ariyawansa and Felt [4]; problems randomly generated by the SLP-IOR system of Kall and Mayer [85]; and real-life gas-purchase planning problems by Koberstein et al. [94].

To see the capacities and scale-up properties of the different methods, we solved each problem many times, with extending scenario sets. Our paper [190] contains detailed computational results, i.e., solution times for every method and every problem instance. (In accord with *Mathematical Programming Computation* editorial policy, all results were checked by the Editors.) Here I only recount statistics and observations that I consider relevant.

Our computational study demonstrates that decomposition methods generally scale better than direct solution of the equivalent LP problem. In particular, the simplex method is very sensitive to increasing scenario numbers. The interior-point method scales much better than the simplex, though the barrier method applied to the equivalent LP problem requires large memory. Concerning solution capacity, the largest two instances of the real-life gas-purchase planning problems resisted direct solution by both the simplex and the barrier solver. (Either numerical difficulties occurred or the time limit of 1800 seconds was exceeded.) In contrast, level decomposition solved all instances with a single exception (time limit was exceeded in case of a medium-sized instance).

Concerning scale-up properties, Figure 4.2 well represents our findings. It shows solution times of the 4node problem with extending scenario sets, cardinalities being 2-powers up to 32768. (All instances were solved to 5 accurate digits in the optimum.)

We found that decomposition based on the aggregate model scales better than that based on the disaggregate model. (The TR and RD methods apply disaggregate models.) This is in agreement with the observation of Birge and Louveaux [10] who conclude that the disaggregate formulation is in general more effective when the number of the scenarios is small. However, we found that the break-even threshold may be high, depending on the solver applied to the master problem.

In our experiments plain decomposition performed well. Using level regularisation we were able to solve very large instances of difficult application problems.

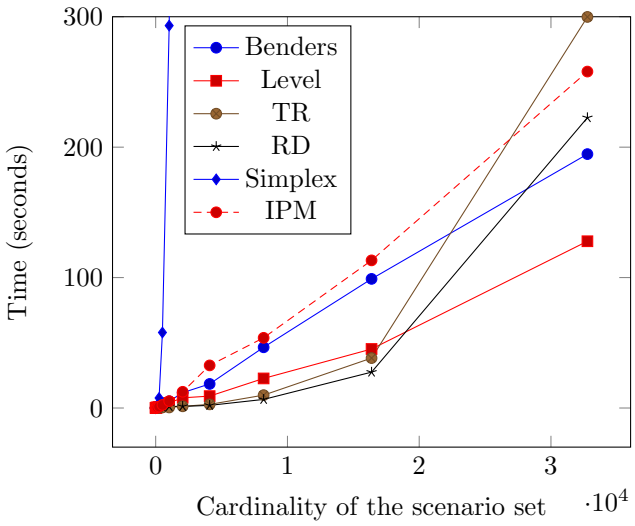

Figure 4.2: Dependece of the solution time on the size of the scenario set. Cardinality (in tens of thousands) is measured on the horizontal axis.

Combining the advantages of the aggregate and the disaggregate model

I specialized the on-demand accuracy approach of de Oliveira and Sagastizábal [27], and applied the resulting partially exact level method to two-stage stochastic programming problems. I showed that this special approach combines the advantages of the traditional aggregate and disaggregate models.

The following algorithm is a specialization of Algorithm 8. The main feature is that the descent target (in step *20.4*, below) is set to $\kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i$, in accordance with Proposition 11. Here the target regulating parameter κ is fixed according to (2.23). Another minor modification is that exact supporting functions are constructed at substantial iterates (i.e., the accuracy tolerance of the oracle is set to 0). — In our papers [64] and [184], the term 'partly inexact' was used for the resulting algorithm. In this dissertation, I call the method 'partially exact' to keep the terminology consistent.

Algorithm 20 *A partially exact level method.*

20.0 Parameter setting.
Set the stopping tolerance $\epsilon > 0$.

Set the level parameter λ $(0 < \lambda < 1)$.
Set the target regulating parameter κ such that $0 < \kappa < 1 - \lambda$.

20.1-3
are the same as the corresponding steps of Algorithm 8.

20.4 Bundle update.

Let $\delta_{i+1} = 0$.

Call an oracle of Specification 9 with the following inputs:
- the current iterate $\boldsymbol{x}_{i+1}$,
- the accuracy tolerance 0, and
- the descent target $\kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i$.

Let $l_{i+1}(\boldsymbol{x})$ be the linear function returned by the oracle.

If the descent target was reached then let $\mathcal{J}_{i+1} = \mathcal{J}_i \cup \{i+1\}$, otherwise let $\mathcal{J}_{i+1} = \mathcal{J}_i$.

Increment i, and repeat from step *20.2*.

We apply the above algorithm to an aggregate master problem, hence we have $\varphi_i(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} + \widetilde{f}(\boldsymbol{x})$. The oracle, on the other hand, will retain all the information that permits the construction of the disaggregate model function $\boldsymbol{c}^T\boldsymbol{x}+\widetilde{q}(\boldsymbol{x})$ – namely, the sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$. – This is just the way proposed in de Oliveira and Sagastizábal [27]. But due to our special target level formulation, the decision in the oracle now takes a very instructive form. If

$$\boldsymbol{c}^T\boldsymbol{x}_{i+1} + \widetilde{q}(\boldsymbol{x}_{i+1}) > \kappa\Big\{\boldsymbol{c}^T\boldsymbol{x}_{i+1} + \widetilde{f}(\boldsymbol{x}_{i+1})\Big\} + (1-\kappa)\overline{\phi}_i \tag{4.8}$$

or, equivalently,

$$\widetilde{q}(\boldsymbol{x}_{i+1}) - \widetilde{f}(\boldsymbol{x}_{i+1}) > \frac{1-\kappa}{\kappa}\left[\,\overline{\phi}_i - \Big(\boldsymbol{c}^T\boldsymbol{x}_{i+1} + \widetilde{q}(\boldsymbol{x}_{i+1})\Big)\,\right] \tag{4.9}$$

holds, then an approximate support function will be created using the information already stored in the oracle. Otherwise, the second-stage problems will be solved. – Of course, here $\boldsymbol{x}_{i+1}$ is the new iterate, and $\overline{\phi}_i$ denotes the best objective value known at the present stage of the solution process.

Justification of the decision rule (4.8). We need an approximate support function $\ell(\boldsymbol{x})$ that satisfies the requirements of the oracle in Algorithm 9, with the input parameters set in *20.4*.

Let $\widetilde{\ell}(\boldsymbol{x})$ denote a linear support function of the disaggregate model function $\boldsymbol{c}^T\boldsymbol{x} + \widetilde{q}(\boldsymbol{x})$ at the new iterate. $\widetilde{\ell}(\boldsymbol{x})$ is easily constructed from the information stored in the oracle.

If (4.8) holds, then $\ell(\boldsymbol{x}) = \widetilde{\ell}(\boldsymbol{x})$ certifies that the descent target cannot be attained. Otherwise $\ell(\boldsymbol{x})$ will be constructed by solving the second-stage problems, and hence $\ell(\boldsymbol{x}_{i+1}) = \boldsymbol{c}^T\boldsymbol{x}_{i+1} + q(\boldsymbol{x}_{i+1})$ will hold.

Assumed that all the dual feasible solutions in the sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$ are basic solutions, a common upper bound exists on the gradients $\|\nabla\ell\|$ constructed in the course of the process. □

For an interpretation of the decision rule in terms of the two-stage problem, let us observe that the left-hand side of (4.9) is the difference between the disaggregate and the aggregate model function values. The expression in the

square bracket on the right-hand side estimates the improvement of the new iterate over the best function value previously known. If the disaggregate model is significantly better than the aggregate one, then the latter will be improved by including information extracted from the former.

We implemented the method and performed an extensive computational study in collaboration with Leena Suhl, Achim Koberstein and Christian Wolf from the DS&OR (Decision Support & Operations Research) Lab of Paderborn University. This is one of the leading industrial optimization teams in Germany. Our joint implementation project was based on the parallel nested Benders solver [185] previously developed by my co-workers at Paderborn University.

Our joint project is discussed in [184]. We evaluated the performance of the following methods:

Level-ODA: level decomposition with on-demand accuracy. Namely, Algorithm 20 with the parameter setting $\lambda = 0.5$, $\kappa = 0.5$.

Level: level decomposition. Similar to Algorithm 20 but the second-stage problems are solved in each iteration. We have set $\lambda = 0.5$.

Benders-SC: plain decomposition based on the aggregate model (also called single-cut method).

Benders-MC: plain decomposition based on the disaggregate model (also called multi-cut method).

Benders-ODA: unregularized decomposition based on the aggregate model, but applying the on-demand accuracy approach. Namely, Algorithm 20 but the regularization is switched off by the extremal setting of $\lambda = 0$. We have set $\kappa = 0.5$.

DEQ: direct solution of the equivalent linear programming problem.

The optimal solution of the expected value problem was chosen as the initial first-stage solution. All experiments were carried out on a processor supporting parallel processing. It had with four physical cores, but eight logical cores due to hyper-threading. The master problem and the recourse problems in the decomposition schemes were solved with the CPLEX dual simplex solver, using one thread each. The equivalent linear programming problems were solved with the CPLEX barrier solver, using eight threads.

Test problems were drawn from the following sources: the test set composed by Deák and used in his computational study [33]; the Slptestset collection of Ariyawansa and Felt [4]; problems randomly generated by the SLP-IOR system of Kall and Mayer [85]; LP relaxations of the sslp problems contributed by Ntaimo and Sen, available in the SIPLIB stochastic integer test set library [3]; the POSTS collection of Holmes [82]; real-life gas-purchase planning problems by Koberstein et al. [94]; and sampled versions of the instances contained in the testset used by Linderoth et al. [101].

Each problem was solved many times, with extending scenario sets. We tested the methods on a total of 105 different problem instances. The solution times are wall-clock times of the solution process, given in seconds. (A time limit of 3,600 seconds was enforced for each run.) Our paper [184] contains detailed computational results, i.e., solution times for every method and every problem instance. Here I only recount statistics and observations that I consider relevant.

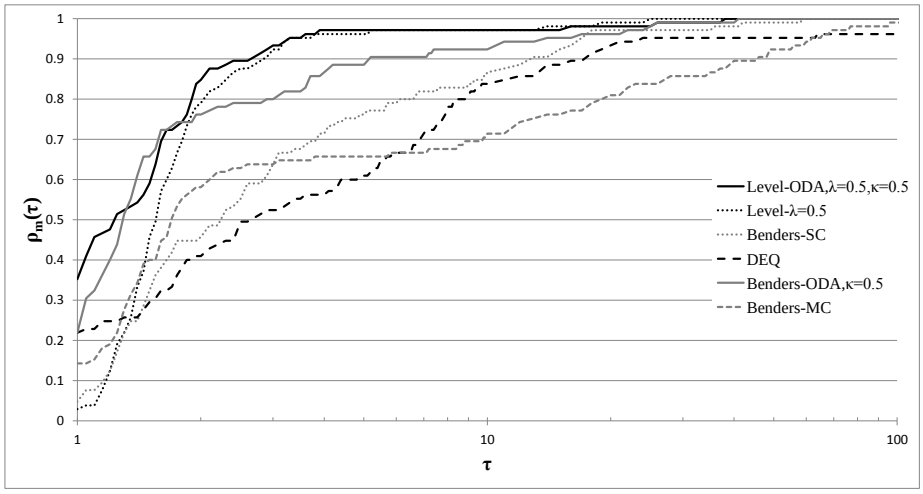

Figure 4.3: Performance profiles of the algorithms.

Performance profiles of the different algorithms are presented in Figure 4.3. Performance profiles are a widely used tool for visual comparison of different solution methods with regard to a given set of test problems. A separate function represents the relative performance of each method. Given a method, the corresponding function value at $\tau \geq 1$ is the percentage of the problems that the given method can solve within a factor τ of the respective fastest methods for these problems. Formal definition of the performance profiles is given in Appendix A.1.

Figure 4.3 shows that level decomposition with on-demand accuracy solved more than a third of the problems faster than the other algorithms. Moreover, it solved about 88% of all problems within a factor of two of the fastest algorithm.

Performance profiles make no distinction between easy and hard problems. In our computational study, regularization and the on-demand accuracy approach had little effect (even adverse effect) on easy problems. But they proved definitely advantageous for hard problems.

Mean computing times are given in Table 4.1. Considering total computing times, level decomposition with on-demand accuracy ranks first. It solved the problems in 21% of the time needed by single-cut Benders. The on-demand accuracy approach without regularization is notably slower than the regularized methods, but still faster than single-cut Benders, taking 45% of the computing

	total	stage 1	stage 2
Level-ODA	11.20	1.40	9.49
Level	14.92	1.28	13.33
Benders-SC	53.27	2.69	50.27
Benders-MC	263.60	258.58	4.79
Benders-ODA	24.08	4.27	19.50
DEQ	72.51		

Table 4.1: Mean computing times. (There is an unsignificant difference between the total time and the sum of stagewise times. This is setup time, typically 0.3.)

time of single-cut Benders.

Multi-cut Benders ranks last in terms of total computing time, taking 495% of the computing time of single-cut Benders. This is because the scale-up properties of multi-cut methods are worse than those of the single-cut methods. It must be noted, however, that much depends on the solver used for the solution of the master problem. (And on the difficulty of the second-stage problems, as compared to the number of the first-stage variables.) Computational efforts spent in solving the first-stage and second-stage problems, respectively, are also presented in Table 4.1. The multi-cut solver spent almost all efforts in the first stage.

Mean iteration counts are shown in Table 4.2. Although multi-cut Benders needs only about 7% of the iterations of single-cut Benders, the computing time is much higher.

The reason why on-demand accuracy speeds up the solution process can be seen by comparing the overall iteration and substantial iteration counts in Table 4.2. (Second-stage problems are only solved in the course of substantial iterations.) Compared with the level method, the on-demand accuracy approach performs five percent more iterations, but 42% less substantial iterations. (The same holds for Benders decomposition with on-demand accuracy, compared with single-cut Benders decomposition: 20% more iterations, but 81% less substantial iterations.)

	total	substantial
Level-ODA	78.83	43.05
Level	74.82	74.82
Benders-SC	274.38	274.38
Benders-MC	20.19	20.19
Benders-ODA	328.89	53.25

Table 4.2: Mean iteration counts in decomposition methods.

Our results are in accordance with the practical efficiency estimate (2.13). This holds for both the level method and the partially exact level method. We

found that there were generally less than n iterations between any two consecutive critical iterations. (n denotes the number of the first-stage variables.) For each problem instance, we considered the maximal number of iterations occurring between any two consecutive critical iterations. This was then divided with the number of the first-stage variables. The ratios fell below 2 in 99 problems from the 105 tested. In the remaining 6 cases, the ratios were below 8. (The corresponding problem instances belonged to the same problem scheme.)

4.4 Application of the results

The decomposition-based solvers developed in collaboration with the Brunel team ([190]) have been included in the FortSP stochastic programming solver system of OptiRisk, an informatics and consulting company specializing in risk management, and utilizing the results of research done at Brunel University. FortSP has been applied in diverse real-life application projects. My contribution is stated in the manual [191]. – One of my co-workers from Brunel University was Victor Zverovich. His PhD dissertation [189] is based to a large extent on our joint project.

At Paderborn University, our results have been applied by Achim Koberstein and Christian Wolf in developing effective solution methods for the strategic gas-purchase planning problems. In their first approach [94], they directly solved the equivalent LP formulations. Influenced by the computational study [190], they developed decomposition methods, as reported in [185]. The aim of our collaboration with the Paderborn team had been the development of even more effective solvers for the gas problem. All the decomposition-based solvers developed in the course of our collaboration ([184]) proved to be superior to direct solution.

	500 scenarios	1000 scenarios
Level-ODA	134.5	239.3
Level	166.1	272.7
Benders-SC	257.1	393.6
Benders-MC	60.8	192.7
Benders-ODA	141.2	249.5
DEQ	799.3	-

Table 4.3: Computing times for gas-purchase planning problem instances. (The general-purpose interior-point solver returned an incorrect solution for the LP-equivalent of the 1000-scenario problem.)

Computing times of the gas problem are shown in Table 4.3. Data confirm the observation that the aggregate approach scales better than the disaggregate one. While moving from 500 to 1000 scenarios, the computing time of the disaggregate solver increased by 220%, but computing times of aggregate solvers increased by only 50-80%. (The multi-cut method is based on the disaggregate

model, while the single-cut and level-type methods are based on the aggregate model). However, the break-even threshold has not been reached by the instances solved in our project. High break-even threshold is the consequence of a special feature of the gas model: the second-stage problems are very hard.

One of my co-workers at Paderborn University was Christian Wolf. He developed a version of the partially exact level method (of Algorithm 20) which surpassed the multi-cut approach already on the 1000-scenario instance of the gas problem. His idea was to use infinity norm in the projection subproblem (2.11), instead of Euclidean norm. The algorithm was described and the results were reported in his PhD thesis [183], written partly with me as advisor.

Our findings published in [190] served as guidelines to J. Gondzio and his team at the University of Edinburgh. They developed a decomposition framework that relies on their primal-dual interior-point solver. They call the approach primal-dual column generation method (PDCGM). Algorithmic descriptions and computational study were published in Gondzio, González-Brevis and Munari [73]. Comparing their findings with ours reported in [190], they observe that their PDCGM implementation is competitive with the solvers we developed in collaboration with the Brunel team. — Enhanced solvers developed in collaboration with the Paderborn team surpassed the PDCGM in efficiency.

Our computational study [190] was considered a reference point by Takano et al. [167] and Sen and Liu [154].

4.5 Summary

I adapted the approximate level method [51] to the aggregate master problem. The (inexact) oracle was based on a successive approximation of the distribution. We implemented the method with my doctoral student Zoltán Szőke at the Eötvös Loránd University. The method was described and a computational study was presented in [62].

To my knowledge, it was the first attempt to regularize the aggregate master problem in a decomposition framework for the two-stage stochastic programming problem. The novelty of the approach is that the accuracy of the distribution approximation is regulated by the optimization method applied to the master problem. This setup helps in finding a balance between different efforts. Our numerical results show that this approximation framework is workable, and that the approximate level method inherits the practical efficiency of the level method [99].

Our work influenced the projects of Oliveira, Sagastizábal and Scheimberg [119] and Song and Luedtke [157].

The motivation of the computational study [190] was to re-assess different solution approaches to two-stage problems, in view of recent advances in computer architecture and LP solver algorithms. My co-workers were Gautam Mitra, Francis Ellison and Victor Zverovich of the CARISMA team (Centre for the

Analysis of Risk and Optimisation Modelling Applications) from Brunel University, London. This was one of the leading teams in computational stochastic programming world-wide.

To see the capacities and scale-up properties of the different solution approaches, we solved each of our test problems many times, with extending scenario sets. Our computational study demonstrates that decomposition methods generally scale better than direct solution of the equivalent LP problem. Moreover, we found that decomposition based on the aggregate model scales better than that based on the disaggregate model. This is in agreement with the observation of Birge and Louveaux [10] who conclude that the disaggregate formulation is in general more effective when the number of the scenarios is small. However, we found that the break-even threshold may be high, depending on the solver applied to the master problem. – Using level regularisation in the aggregate master problem, we were able to solve very large instances of difficult application problems.

The decomposition-based solvers developed in collaboration with the Brunel team have been included in the FortSP stochastic programming solver system of OptiRisk, an informatics and consulting company specializing in risk management, and utilizing the results of research done at Brunel University. FortSP has been applied in diverse real-life application projects. My contribution is stated in the manual [191].

One of my co-workers from Brunel University was Victor Zverovich. His PhD dissertation [189] is based to a large extent on our joint project.

Our computational study influenced the team of A. Koberstein at Paderborn University to develop decomposition methods to solve their application problems, as reported in [185].

Our findings served as guidelines to J. Gondzio and his team at the University of Edinburgh in their project [73], and were considered as reference by Takano et al. [167] and Sen and Liu [154].

I specialized the on-demand accuracy approach of [27], and applied the resulting partially exact level method to two-stage stochastic programming problems. I showed that the partially exact level method admits a special formulation of the descent target. Using that, the decision in the oracle takes an instructive form: no recourse problem is solved if the disaggregate model function value is significantly higher than the aggregate one, as evaluated at the new iterate. Hence the partially exact level method with this special descent target combines the advantages of the traditional aggregate and disaggregate models.

We implemented the method and performed an extensive computational study in collaboration with Leena Suhl, Achim Koberstein and Christian Wolf from the DS&OR (Decision Support & Operations Research) Lab of Paderborn University. This is one of the leading industrial optimization teams in Germany.

We tested the methods on a total of 105 different problem instances. Each problem was solved many times, with extending scenario sets. The new method solved more than a third of the problems faster than the other algorithms. Moreover, it solved about 88% of all problems within a factor of two of the

fastest algorithm. Considering total computing times, the new method also ranked first. It solved the problems in 21% of the time needed by single-cut Benders that we considered the benchmark algorithm. Test results demonstrate that the partially exact level method inherits the practical efficiency of the level method [99].

The methods and solvers developed in the course of our collaboration with the Paderborn team have been applied in the effective solution of the strategic gas-purchase planning problem, a real-life project of theirs.

One of my co-workers at Paderborn University was Christian Wolf. He developed a version of the partially exact level method which proved the best method on the 1000-scenario instance of the gas problem. The algorithm was described and the results were reported in his PhD thesis [183], written partly with me as advisor.

Chapter 5

Feasibility issues in two-stage stochastic programming problems

In this chapter I discuss ways of dealing with second-stage infeasibility. We work with the two-stage stochastic programming problem of Chapter 4, but drop Assumption 19 on relatively complete recourse.

Accordingly, let K_s $(s = 1, \ldots, S)$ denote the set of those $\boldsymbol{x}$ vectors for which $\mathcal{R}_s(\boldsymbol{x})$ is feasible. The domain of the recourse function $q_s(\boldsymbol{x})$ is K_s, and the domain of the expected recourse function $q(\boldsymbol{x})$ is $K = K_1 \cap \cdots \cap K_S$. We assume that $X \cap K$ is not empty. In the first-stage problem formulation (4.2), the constraint $\boldsymbol{x} \in K$ must be added to ensure second-stage feasibility. This type of constraint is called *induced constraint*.

According to linear programming duality, the recourse problem $\mathcal{R}_s(\boldsymbol{x})$ is feasible if and only if the dual recourse problem $\mathcal{D}_s(\boldsymbol{x})$ has a finite optimum. This allows a characterization of K_s using feasible rays of the dual recorse problem; namely, $\boldsymbol{x} \in K_s$ if and only if

$$\boldsymbol{v}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) \leq 0 \quad \text{holds with any ray } \boldsymbol{v}_s \text{ of the feasible domain of } \mathcal{D}_s(\boldsymbol{x}).$$

Given a finite subset $\widetilde{V}_s$ of the rays of the feasible domain of $\mathcal{D}_s(\boldsymbol{x})$, the corresponding cuts define

$$\widetilde{K}_s = \{\, \boldsymbol{x} \mid \boldsymbol{v}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) \leq 0 \;\; (\boldsymbol{v}_s \in \widetilde{V}_s) \,\}. \tag{5.1}$$

This is an outer approximation of K_s.

The cuts defining $\widetilde{K}_s$ are then included in the model problems (4.5) and (4.6).

5.1 Historical perspectives

A decomposition framework is easily extended to handle induced constraints, and the cutting-plane viewpoint admits a clear visual image of the corresponding cuts. The first decomposition method capable of handling induced constraints was the L-shaped method of Van Slyke and Wets [176].

Given an iterate $\widehat{\boldsymbol{x}}$, if all the recourse problems $\mathcal{R}_s(\widehat{\boldsymbol{x}})$ $(s = 1, \ldots, S)$ are feasible, then new cuts are added to the respective model functions (4.4), or a single cut is added to the aggregate model function (4.7). These are called *optimality cuts* in this context. If, on the other hand, the recourse problem $\mathcal{R}_s(\widehat{\boldsymbol{x}})$ is not feasible for some s, then, solving $\mathcal{D}_s(\widehat{\boldsymbol{x}})$ by a simplex-type method, we obtain a ray such that the corresponding cut is violated by $\widehat{\boldsymbol{x}}$. The appropriate cut is then added to the model (5.1). These are called *feasibility cuts* in this context.

Ruszczyński [150] admits feasibility cuts in his regularized decomposition method. The drawback is that regularization does not extend to feasibility cuts. The scope of optimization may alternate between minimizing the objective function and satisfying feasibility cuts.

An accelerated version of the regularized decomposition method was developed by Ruszczyński and Świętanowski [152]. One of the enhancements is a penalized formulation of the recourse problems. The penalty parameter is adjusted as the solution procedure progresses. The aim is to facilitate crash and warm starts, and to allow freedom in model formulation. Second-stage feasibility is still ensured by adding feasibility cuts in the first-stage problem.

Prékopa in [132], Chapter 12.10 introduced non-negative slack variables in the recourse problems, and observed that we get an extension of the recourse function if the penalties are large enough, but he did not examine the necessary magnitude.

5.2 Contribution

In [62], I proposed handling second-stage infeasibility through a constraint function in the master problem, and adapted the approximate constrained level method of [51], recounted in Chapter 2, to solve the resulting special master problem. This approach avoids feasibility cuts, and the regularization extends to feasibility issues.

But this approach requires extending the recourse function to the whole space. I worked with penalized formulations of the recourse problem, and examined the necessary magnitude of the penalty.

We implemented the method with Zoltán Szőke who was my doctoral student at the Eötvös Loránd University.

Problem formulation

Let us first extend the recourse problems of (4.1). We add a slack vector $\boldsymbol{d}$ in the primal problem, and penalize its norm with a positive weight $w \in \mathbb{R}$. The resulting problem is $\mathcal{R}_s(\boldsymbol{x}, w)$, below.

$$\mathcal{R}_s(\boldsymbol{x}, w) \quad \begin{array}{l} \min \boldsymbol{q}_s^T \boldsymbol{y} + w\|\boldsymbol{d}\|_\diamond \\ \text{such that} \\ T_s\boldsymbol{x} + W_s\boldsymbol{y} + \boldsymbol{d} = \boldsymbol{h}_s, \\ \boldsymbol{y} \geq \boldsymbol{0}, \end{array} \quad \left| \quad \mathcal{D}_s(\boldsymbol{x}, w) \quad \begin{array}{l} \max \boldsymbol{z}^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) \\ \text{such that} \\ W_s^T\boldsymbol{z} \leq \boldsymbol{q_s}, \\ \|\boldsymbol{z}\|_\square \leq w. \end{array} \right. \tag{5.2}$$

In order to retain linearity, we may use either $\|\boldsymbol{d}\|_\diamond = \|\boldsymbol{d}\|_1$ or $\|\boldsymbol{d}\|_\diamond = \|\boldsymbol{d}\|_{\max}$. Keeping to linear programming formulations, the dual of problem $\mathcal{R}_s(\boldsymbol{x}, w)$ takes the form of $\mathcal{D}_s(\boldsymbol{x}, w)$. The norm $\| \ \|_\square$ in $\mathcal{D}_s(\boldsymbol{x}, w)$ depends on the selection of the norm $\| \ \|_\diamond$ in $\mathcal{R}_s(\boldsymbol{x}, w)$. For $\| \ \|_\diamond = \| \ \|_1$ we have $\| \ \|_\square = \| \ \|_{\max}$ and vice versa. – I keep to $\|\boldsymbol{d}\|_\diamond = \|\boldsymbol{d}\|_1$ for the sake of simplicity.

Problem $\mathcal{R}_s(\boldsymbol{x}, w)$ is feasible for any $(\boldsymbol{x}, w)$. Let $q_s(\boldsymbol{x}, w)$ denote the optimal objective value (possibly $-\infty$). This is increasing in w. I show that the objective function is bounded if w is large enough: Let $\boldsymbol{z}_s$ denote a feasible solution of $\mathcal{D}_s(\boldsymbol{x})$ that we have assumed to exist. Let $\underline{w}_s := \|\boldsymbol{z}_s\|_\square$. If we have $w \geq \underline{w}_s$ then $\boldsymbol{z}_s$ is a feasible solution of $\mathcal{D}_s(\boldsymbol{x}, w)$, establishing a lower bound for the objective value of $\mathcal{R}_s(\boldsymbol{x}, w)$. This is easily computable. In what follows we assume that $w \geq \underline{w}_s$ $(1 \leq s \leq S)$ holds.

$q_s(\boldsymbol{x}, w)$ is a polyhedral convex function, and $q_s(\boldsymbol{x}, w) \leq q_s(\boldsymbol{x})$ holds for any $\boldsymbol{x} \in K_s$. Hence the expectation $q(\boldsymbol{x}, w) = \sum_{s=1}^S p_s q_s(\boldsymbol{x}, w)$ is a convex function, and $q(\boldsymbol{x}, w) \leq q(\boldsymbol{x})$ holds for any $\boldsymbol{x} \in K$.

Let us measure the infeasibility of the recourse problem $\mathcal{R}_s(\boldsymbol{x})$ by the function

$$g_s(\boldsymbol{x}) = \begin{array}{l} \min \|\boldsymbol{d}\|_\diamond \\ \text{such that} \\ T_s\boldsymbol{x} + W_s\boldsymbol{y} + \boldsymbol{d} = \boldsymbol{h}_s, \\ \boldsymbol{y} \geq \boldsymbol{0}. \end{array} \tag{5.3}$$

This problem has an optimal solution for any $\boldsymbol{x}$, and we have $g_s(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} \in K_s$. Let $g(\boldsymbol{x}) = \sum_{s=1}^S p_s g_s(\boldsymbol{x})$ denote the expectation of the inconsistency measure. This is a convex function and $g(\boldsymbol{x}) \leq 0$ holds if and only if $\boldsymbol{x} \in K$.

Assumption 21 *The weight w has been set in such a way that $q(\boldsymbol{x}, w) = q(\boldsymbol{x})$ holds for $\boldsymbol{x} \in K$.*

Under Assumption 21, the first-stage problem can be written in the constrained convex programming form

$$\min \ \boldsymbol{c^T x} + q(\boldsymbol{x}) \quad \text{such that} \quad \boldsymbol{x} \in X \ \text{ and } \ g(\boldsymbol{x}) \leq 0. \tag{5.4}$$

This formulation admits more effective solution methods than the application of feasibility cuts. The approximate constrained level method that I adapted to this problem is a primal-dual method, and the rule of tuning the dual iterate keeps a fine balance between feasibility and optimality issues.

Determining the penalty

Let us now examine the necessary magnitude of the weight which ensures that Assumption 21 holds. Let

$$\overline{w}_s = \max \|\boldsymbol{z}\|_{\square} \tag{5.5}$$

while $\boldsymbol{z}$ sweeps through the set of the feasible basic solutions of the system $W_s^T \boldsymbol{z} \leq \boldsymbol{q}_s$, i.e., of the constraint set of problem $\mathcal{D}_s(\boldsymbol{x})$. (Basic solutions are defined by first converting the system into an equality system by adding slack variables. – We assumed that the dual recourse problems are feasible.) Let moreover $\overline{w} = \max_{1 \leq s \leq S} \overline{w}_s$.

Observation 22 *Assumption 21 holds if we have $w \geq \overline{w}$.*

Proof. Given scenario s and $\boldsymbol{x} \in K_s$, the problem $\mathcal{D}_s(\boldsymbol{x})$ has an optimal solution. Hence an optimal basic solution also exists, let $\boldsymbol{z}_s^\star$ denote one of those.

Since we have $\|\boldsymbol{z}_s^\star\|_{\square} \leq \overline{w}$ by the definition of $\overline{w}$, it follows that $\boldsymbol{z}_s^\star$ is a feasible solution of $\mathcal{D}_s(\boldsymbol{x}, w)$ provided $w \geq \overline{w}$. □

If $\overline{w}$ is not known, then the penalty parameter can be adjusted in the course of the solution process, ensuring that $q(\boldsymbol{x}_i, w) = q(\boldsymbol{x}_i)$ holds for the known iterates; in accordance with the idea of Ruszczyński and Świętanowski [152]. I have worked out the necessary adjustments in the approximate constrained level method. We implemented this adjustment scheme with Zoltán Szőke, and reasonably low weights proved sufficient in our experiments.

There are special problem classes where the bound $\overline{w}$ is computable, the most notable is that of network recourse problems.

Observation 23 *If the recourse problems $\mathcal{R}_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$ are network flow problems, then $\overline{w} \leq \max_{1 \leq s \leq S} \|\boldsymbol{q}_s\|_1$.*

Proof easily follows from the construction of the dual vector in the network simplex method, based on the observation of Fulkerson and Dantzig [67] that any basis matrix in a network problem corresponds to a rooted spanning tree. □

With the growing size of network design problems that need to be solved in logistics and transportation, my approach is worthwhile to examine in a practical framework, as observed by Rahmaniani, Crainic, Gendreau and Rei [138].

A technical comment

The price we pay for applying the constrained formulation (21) is that two recourse problems need to be solved for every scenario in each master iteration: $\mathcal{R}_s(\boldsymbol{x}, w)$ to evaluate the recourse function, and (5.3) to measure infeasibility. Though it turns out that the solution of a single second-stage problem is often sufficient.

Observation 24 *Let $w > \overline{w}$. Then, for each scenario s and $\boldsymbol{x} \in X$, an optimal basic solution of problem $\mathcal{R}_s(\boldsymbol{x}, w)$ is also an optimal solution of problem (5.3).*

The proof is based on the well-know relationship between primal and dual LP problems that I recount in Appendix A.2.

Proof. Let $\widehat{\boldsymbol{x}} \in X$. Let us consider a scenario s and let $\widehat{w} > \overline{w}_s$. The definition (5.5) of $\overline{w}_s$ ensures that any feasible basis of the problem $\mathcal{D}_s(\widehat{\boldsymbol{x}}, \widehat{w})$ must contain all the slack vectors corresponding to the constraints $\|\boldsymbol{z}\|_\square \leq w$. The rest of the basis vectors form a basis of problem $\mathcal{D}_s(\widehat{\boldsymbol{x}})$. Such a basis of $\mathcal{D}_s(\widehat{\boldsymbol{x}}, \widehat{w})$ remains feasible for any $w > \overline{w}_s$. It follows that each of the dual recourse problems $\{\, \mathcal{D}_s(\widehat{\boldsymbol{x}}, w) \,|\, w > \overline{w}_s \,\}$ has the same set of feasible bases.

According to Observation 57 in Appendix A.2, each of the recourse problems $\{\, \mathcal{R}_s(\widehat{\boldsymbol{x}}, w) \,|\, w > \overline{w}_s \,\}$ has the same set of dual feasible bases. Let $\widehat{\mathcal{B}}$ be an optimal basis (that we assumed to exist), and let $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{d}})$ denote the corresponding optimal basic solution. Given a feasible solution $(\boldsymbol{y}, \boldsymbol{d})$,

$$\boldsymbol{q}_s^T \boldsymbol{y} - \boldsymbol{q}_s^T \widehat{\boldsymbol{y}} \;\geq\; w\big(\|\widehat{\boldsymbol{d}}\|_\diamond - \|\boldsymbol{d}\|_\diamond \big) \quad \text{holds for any} \quad w > \overline{w}_s,$$

due to the optimality of $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{d}})$. The left-hand side does not depend on w, hence the right-hand side must be non-positive. It follows that $(\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{d}})$ is an optimal solution of the problem (5.3: $\boldsymbol{x} = \widehat{\boldsymbol{x}}$). □

5.3 Summary

In [62], I proposed handling second-stage infeasibility through a constraint function in the master problem, and adapted the approximate constrained level method [51] to solve the resulting special master problem.

This formulation admits more effective solution methods than the application of feasibility cuts. The approximate constrained level method that I adapted to this problem is a primal-dual method, and the rule of tuning the dual iterate keeps a fine balance between feasibility and optimality issues. Moreover, the regularization extends to feasibility issues.

But this approach requires extending the recourse function to the whole space. I worked with penalized formulations of the recourse problem, and examined the necessary magnitude of the penalty. There are special problem classes where the necessary magnitude is computable, the most notable is that of network recourse problems. For general recourse problems, I worked out a means of adjusting the penalty parameter in the course of the solution process.

We implemented the methods with Zoltán Szőke who was my doctoral student at the Eötvös Loránd University. Computational experiments confirmed the workability of the approach.

With the growing size of network design problems that need to be solved in logistics and transportation, this approach is worthwhile to examine in a practical framework, as observed by Rahmaniani, Crainic, Gendreau and Rei [138].

Chapter 6

Risk constraints in two-stage stochastic programming problems

In this chapter I discuss solution schemes for two-stage stochastic programming problems with CVaR and stochastic ordering constraints applied on the recourse function. We work with the two-stage stochastic programming problem of Chapter 4, keeping Assumption 19 on relatively complete recourse.

6.1 Background

Given $\boldsymbol{x} \in X$ let us consider the recourse function values $q_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$ as realizations of a random recourse function value $Q(\boldsymbol{x})$.

A risk constraint on the recourse. Ahmed [1] adds a risk constraint in the form $\mathcal{G}\left(Q(\boldsymbol{x})\right) \leq \rho$ to the first-stage problem (4.2). The function $\mathcal{G}$ maps a certain family of random variables to the set of the real numbers, and ρ is a constant.

A coherent risk measure $\mathcal{G}$ results in a convex $\boldsymbol{x} \mapsto \mathcal{G}\left(Q(\boldsymbol{x})\right)$ function. Among other risk mappings, Ahmed considers applying CVaR in the role of $\mathcal{G}$. For the solution of the resulting problems, he develops special cutting plane methods. One type uses disaggregate cuts. The other type uses aggregate cuts, and employs parametric programming to explore the efficient frontier.

The first-stage problem takes the form

$$\begin{array}{rl} \min & \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x} + \mathrm{E}\left(Q(\boldsymbol{x})\right) \\ \text{such that} & \boldsymbol{x} \in X, \\ & \beta\,\mathrm{CVaR}_{\beta}\left(Q(\boldsymbol{x})\right) \leq \rho, \end{array} \tag{6.1}$$

where the parameters β and ρ are set by the decision maker.

A stochastic ordering constraint on the recourse. Two-stage problems with stochastic ordering constraints were first considered by Schultz and co-workers, an overview can be found in [46].

Dentcheva and Martinez [38] impose an increasing convex ordering constraint on the recourse. Based on Theorem 1.5.7 in [111], they adopt the following characterization as a definition of the increasing convex order: given integrable random variables $Q, \widehat{Q}$ representing costs, $Q \preceq_{_{IC}} \widehat{Q}$ if and only if

$$\mathrm{E}(\,[Q-t]_+) \leq \mathrm{E}\big(\,\big[\widehat{Q}-t\big]_+\big) \quad \text{holds for each } t \in \mathbb{R}.$$

This is analogous to the characterization *(b)* in Chapter 3 of the second-order stochastic dominance relation. We have $Q \preceq_{_{IC}} \widehat{Q}$ if and only if $-Q \succeq_{_{SSD}} -\widehat{Q}$ holds.

Introducing an IC-constraint, the first-stage problem takes the form

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \mathrm{E}\,(Q(\boldsymbol{x})) \\ \text{such that} & \boldsymbol{x} \in X, \\ & Q(\boldsymbol{x}) \preceq_{_{IC}} \widehat{Q}. \end{array} \tag{6.2}$$

Here $\widehat{Q}$ is a given integrable random variable, representing a benchmark cost or loss. Applying results of Ogryczak and Ruszczyński [118], and Dentcheva and Ruszczyński [43], Dentcheva and Martinez develop new characterizations of the increasing convex order. They also construct further finite linear models, based on the results of Dentcheva and Ruszczyński cited in Chapter 3. Dentcheva and Martinez develop two special decomposition methods for the solution of the resulting problems: one uses quantile functions, and the other uses excess functions. These decomposition methods are based on disaggregate models, though the latter also employs aggregate cuts. The authors implemented these methods and present encouraging test results.

6.2 Contribution and application of the results

In [64], I generalized the on-demand accuracy approach to the CVaR-constrained problem (6.1) and the stochastic ordering-constrained problem (6.2).

The proposed method applies the following algorithm that is a specialization of Algorithm 13. The main feature is that the descent target is set according to Corollary 17. Here the target regulating parameter κ is fixed according to (2.23). A minor modification is that exact supporting functions are constructed at substantial iterates (i.e., the accuracy tolerance of the oracle is set to 0).

Algorithm 25 *A partially exact version of the constrained level method.*

25.0 Parameter setting.

Set the stopping tolerance $\epsilon > 0$.

Set the parameters λ and μ $(0 < \lambda, \mu < 1)$.

Set the accuracy regulating parameter κ such that $0 < \kappa < 1 - \lambda$.

25.1-4
are the same as the corresponding steps in Algorithm 13.

25.5 Bundle update.
Let $\delta_{i+1} = 0$.

Call an oracle of Specification 14 with the following inputs:
- the current iterate $\boldsymbol{x}_{i+1}$,
- the current dual iterate α_i,
- the accuracy tolerance 0, and
- the descent target $\kappa\big(\alpha_i\varphi_i(\boldsymbol{x}_{i+1}) + (1-\alpha_i)\psi_i(\boldsymbol{x}_{i+1})\big) + (1-\kappa)\overline{\phi}_i$.

Let $l_{i+1}(\boldsymbol{x})$ and $l'_{i+1}(\boldsymbol{x})$ be the linear functions returned by the oracle.

If the descent target was reached then let $\mathcal{J}_{i+1} = \mathcal{J}_i \cup \{i+1\}$,
otherwise let $\mathcal{J}_{i+1} = \mathcal{J}_i$.

Increment i, and repeat from step *25.2*.

(In [64], the term 'partly inexact' was used for the above algorithm. In this dissertation, I call the method 'partially exact' to keep the terminology consistent.)

Handling a CVaR constraint on the recourse

In order to apply the partially exact constrained level method to problem (6.1), we need an appropriate oracle, i.e., one that satisfies Specification 14. The objective function is $\boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x} + \mathrm{E}\,(Q(\boldsymbol{x}))$, and its handling was described in Chapter 4. The constraint function is $\beta\,\mathrm{CVaR}_\beta\,(Q(\boldsymbol{x})) - \rho$. Direct substitution of the computational formula (3.3) into the CVaR constraint function would lead to an unbounded domain, causing technical problems in the application of a level-type solution method. In the present case, though, we only need supporting linear functions to the constraint function. Applying (3.9), we get that

$$\beta\ \mathrm{CVaR}_\beta\big(Q(\boldsymbol{x})\big) = \max_{(\pi_1,\ldots,\pi_S)\in\Pi} \sum_{s=1}^{S} \pi_s\, q_s(\boldsymbol{x}) \qquad \text{holds for any } \boldsymbol{x}, \tag{6.3}$$

where $\Pi = \big\{\,(\pi_1,\ldots,\pi_S) \in \mathbb{R}^S \;\big|\; 0 \le \pi_s \le p_s\ (s=1,\ldots,S),\ \ \sum_{s=1}^{S}\pi_s = \beta\,\big\}$. (6.3) shows moreover that the constraint function inherits Lipschitz continuity from the recourse functions.

Having fixed $\boldsymbol{x} = \hat{\boldsymbol{x}}$, an optimal solution vector $(\hat{\pi}_1,\ldots,\hat{\pi}_S)$ of (6.3) can be found by just sorting the values $q_s(\hat{\boldsymbol{x}})$ $(s=1,\ldots,S)$. Let $\hat{\ell}_s(\boldsymbol{x})$ $(s=1,\ldots,S)$ denote supporting linear functions to the respective recourse functions $q_s(\boldsymbol{x})$, at $\hat{\boldsymbol{x}}$. Then $\sum_{s=1}^{S}\hat{\pi}_s\,\hat{\ell}_s(\boldsymbol{x})$ is a supporting linear function to $\beta\ \mathrm{CVaR}_\beta\big(Q(\boldsymbol{x})\big)$ at $\hat{\boldsymbol{x}}$, because $\beta\ \mathrm{CVaR}_\beta\big(Q(\boldsymbol{x})\big) \ge \sum_{s=1}^{S}\hat{\pi}_s\,q_s(\boldsymbol{x})$ holds for any $\boldsymbol{x}$, due to (6.3).

As in Chapter 4, let $\widetilde{U}_s$ denote the set of the known dual feasible solutions of the sth recourse problem $(s=1,\ldots,S)$. These sets are maintained in the oracle. A disaggregate model of the function $\mathrm{CVaR}_\beta\big(Q(\boldsymbol{x})\big)$ can be computed as

$\mathrm{CVaR}_\beta\big(\widetilde{Q}(\boldsymbol{x})\big)$, where $\widetilde{Q}(\boldsymbol{x})$ denotes a random function value with realizations $\widetilde{q}_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$. Of course we have $\mathrm{CVaR}_\beta\big(\widetilde{Q}(\boldsymbol{x})\big) \leq \mathrm{CVaR}_\beta\big(Q(\boldsymbol{x})\big)$ due to the monotonicity of CVaR.

Algorithm 26 *A partially exact oracle for the solution of problem (6.1).*

The input parameters are:
$\hat{\boldsymbol{x}}$: the current iterate,
$\hat{\alpha}$: the current dual iterate, and
$\hat{\theta}$: the descent target.
(Concerning accurcy tolerance, $\hat{\delta} = 0$ is assumed.)

Evaluating the disaggregate model of the objective function.

Let $\hat{\boldsymbol{u}}_s$ $(s = 1, \ldots, S)$ be respective optimal solutions of
max $\boldsymbol{u}_s^T(\boldsymbol{h}_s - T_s\hat{\boldsymbol{x}})$ such that $\boldsymbol{u}_s \in \widetilde{U}_s$.

Let $\ell(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} + \sum_{s=1}^S p_s\,\hat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x})$
(a support function to $\boldsymbol{c}^T\boldsymbol{x} + \widetilde{q}(\boldsymbol{x})$ at $\hat{\boldsymbol{x}}$).

Evaluating the disaggregate model of the constraint function.

Let $(\hat{\pi}_1, \ldots, \hat{\pi}_S)$ denote an optimal solution of
max $\sum_{s=1}^S \pi_s\,\widetilde{q}_s(\hat{\boldsymbol{x}})$ such that $(\pi_1, \ldots, \pi_S) \in \Pi$.

Let $\ell'(\boldsymbol{x}) = \sum_{s=1}^S \hat{\pi}_s\,\hat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) - \rho$
(a support function to $\beta\mathrm{CVaR}_\beta\big(\widetilde{Q}(\boldsymbol{x})\big) - \rho$ at $\hat{\boldsymbol{x}}$).

If $\hat{\alpha}\ell(\hat{\boldsymbol{x}}) + (1 - \hat{\alpha})\ell'(\hat{\boldsymbol{x}}) \geq \hat{\theta}$ then *the descent target has not been reached;*

the oracle returns the linear functions $\ell(\boldsymbol{x})$ and $\ell'(\boldsymbol{x})$.

Otherwise *exact supporting functions are constructed:*

Let $\hat{\boldsymbol{u}}_s$ be respective optimal solution of $\mathcal{D}_s(\hat{\boldsymbol{x}})$ $(s = 1, \ldots, S)$, and
let $\ell(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} + \sum_{s=1}^S p_s\,\hat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x})$.

Let $(\hat{\pi}_1, \ldots, \hat{\pi}_S)$ denote the maximizer of (6.3), and
let $\ell'(\boldsymbol{x}) = \sum_{s=1}^S \hat{\pi}_s\,\hat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) - \rho$.

The dual vectors $\hat{\boldsymbol{u}}_s$ $(s = 1, \ldots, S)$ are added to the respective sets $\widetilde{U}_s$, and
the oracle returns the linear functions $\ell(\boldsymbol{x})$ and $\ell'(\boldsymbol{x})$.

(This oracle requires initialization, i.e., the setting of the starting sets $\widetilde{U}_s$ for $s = 1, \ldots, S$.)

Handling a stochastic ordering constraint

I proposed the application of an IC-measure, analogous to the dominance measure (3.11). Let

$$\mathcal{H}\left(Q(\boldsymbol{x})\right) = \min\left\{\xi \in \mathbb{R} \;\middle|\; Q(\boldsymbol{x}) \preceq_{_{IC}} \widehat{Q} + \xi\right\}, \tag{6.4}$$

a function of $\boldsymbol{x}$. Here ξ is a 'certain' (i.e., non-random) loss. Clearly $Q(\boldsymbol{x}) \preceq_{_{IC}} \widehat{Q}$ holds if and only if $\mathcal{H}\left(Q(\boldsymbol{x})\right) \leq 0$. Hence the IC-constrained problem (6.2) can be formulated as

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \mathrm{E}\left(Q(\boldsymbol{x})\right) \\ \text{such that} & \boldsymbol{x} \in X, \\ & \mathcal{H}\left(Q(\boldsymbol{x})\right) \leq 0. \end{array} \tag{6.5}$$

It is easily seen that $\mathcal{H}\left(Q(\boldsymbol{x})\right)$ is a convex function of $\boldsymbol{x}$. Moreover, considering $Q(\boldsymbol{x}) \preceq_{_{IC}} \widehat{Q} + \xi$ represented as a finite system of linear inequalities, a supporting linear function to $\mathcal{H}\left(Q(\boldsymbol{x})\right)$ can be constructed for a given $\widehat{\boldsymbol{x}}$. This allows constructing a convex polyhedral model of the function $\mathcal{H}\left(Q(\boldsymbol{x})\right)$, to be used in the optimization process. For the sake of simplicity, I sketch the construction in the equiprobable case. Converting the relation $Q(\boldsymbol{x}) \preceq_{_{IC}} \widehat{Q} + \xi$ in (6.4) to $-Q(\boldsymbol{x}) \succeq_{_{SSD}} -\widehat{Q} - \xi$, and expressing the latter using tail expectations in the manner of (3.12), we compute $\mathcal{H}\left(Q(\boldsymbol{x})\right)$ as

$$\begin{array}{l} \min\ \xi \\ \\ \text{such that} \\ \\ \mathrm{Tail}_\beta(-Q(\boldsymbol{x})) \;\geq\; \mathrm{Tail}_\beta(-\widehat{Q}) \;-\beta\xi \quad \text{holds for } \beta \in \left\{\tfrac{1}{S}, \tfrac{2}{S}, \ldots, 1\right\}. \end{array} \tag{6.6}$$

Taking into account (3.2), the tail-inequalities can be transformed into CVaR-inequalities. Further simple transformations show that $\mathcal{H}\left(Q(\boldsymbol{x})\right)$ is the upper cover of the functions

$$\mathrm{CVaR}_\beta(Q(\boldsymbol{x})) - \mathrm{CVaR}_\beta(\widehat{Q}) \qquad \left(\beta \in \left\{\frac{1}{S}, \frac{2}{S}, \ldots, 1\right\}\right). \tag{6.7}$$

Polyhedral models of these individual functions can be constructed using (6.3).

In the master problem, we include an aggregate model of $\mathcal{H}\left(Q(\boldsymbol{x})\right)$. In the oracle, on the other hand, we store the results of all the second-stage problems solved. This allows the application of the on-demand accuracy approach.

A computational study

In collaboration with Leena Suhl, Achim Koberstein and Christian Wolf from the DS&OR (Decision Support & Operations Research) Lab of Paderborn University, we implemented and compared different methods for the solution of the CVaR-constrained problem (6.1), and performed an extensive computational study. The following methods were compared:

DEQ: solution of the equivalent linear programming problem, composed using the linear programming formulation of the CVaR constraint, obtained from (3.3).

Benders-Risk: a pure cutting-plane method applied to the aggregate master problem. A special stopping criterion is used: the current gap is computed as the maximum of the dual function h defined as in (2.30), setting $\delta = 0$ always.

Benders-Risk-ODA: an unregularized method with an oracle of on-demand accuracy. The master problem is in aggregate form, but the oracle stores disaggregate information. A dual variable is used to construct a composite function which, in turn, is used to decide whether the second-stage problems need to be solved in the current iteration. (This is the sole role of the dual variable, otherwise the method is not a primal-dual method.) The dual variable is computed as the maximizer of the dual function h defined as in (2.30), setting $\delta = 0$ always. The current gap is computed as the maximum of the dual function.

Level-Risk: the constrained level method of [99]. The aggregate model is used.

Level-Risk-ODA: the partially exact constrained level method of Algorithm 25, with the oracle of Algorithm 26.

Our implementation is based on the solver code described in Wolf and Koberstein [185], Wolf [183] and Wolf, Fábián, Koberstein and Suhl [184]. The implementation can also handle feasibility cuts, in an unregularized manner.

Test problems were drawn from the following sources: the Slptestset collection of Ariyawansa and Felt [4]; problems randomly generated by the SLP-IOR system of Kall and Mayer [85]; the POSTS collection of Holmes [82]; sampled versions of the instances contained in the testset used by Linderoth et al. [101]; and real-life gas-purchase planning problems by Koberstein et al. [94]. We tested the methods on a total of 44 problems.

The expected value problem solution was chosen as the first stage initial solution. We set $\lambda = 0.5$ and $\kappa = 0.5$ for all experiments. The probability β was set to 0.1 in our CVaR formulas, meaning a confidence level 0.9. To set the value for ρ in the CVaR constraint, each test instance was solved both optimizing the expected value and the CVaR of the objective function. Let HN_CVaR be the CVaR of the optimal expected value solution and let MIN_CVaR be the minimized CVaR value. We set ρ to $\frac{1}{2}HN\_CVaR + \frac{1}{2}MIN\_CVaR$, thus guaranteeing that the resulting problem instance is solvable.

All the computing times reported are wall-clock times of the solution process, given in seconds, without the times for reading in the SMPS files. All experiments were carried out on a processor with four physical cores, but eight logical cores due to hyper-threading. The underlying LP solver was the Cplex 12.4 dual simplex solver, with one thread. The Cplex barrier solver was used to solve the equivalent linear programming problems, with eight threads. The

appendix of our paper [64] contains detailed computational results, i.e., solution times for every method and every problem instance. Here I only recount statistics and observations that I consider relevant.

		% of Benders-Risk
DEQ	290	108 %
Benders-Risk	267	100 %
Benders-Risk-ODA	168	63 %
Level-Risk	50	19 %
Level-Risk-ODA	49	18 %

Table 6.1: Average computing times of the different methods.

Table 6.1 shows average computing times of the different methods. The columns contain solution times in mean values and as percentages of those of the Benders-Risk method.

	all iterations	substantial	insubstantial
Benders-Risk	523.1	523.1	
Benders-Risk-ODA	937.9	114.4	823.5
Level-Risk	156.6	156.6	
Level-Risk-ODA	231.2	72.1	159.1

Table 6.2: Averages of master iteration counts of the decomposition methods.

Table 6.2 shows average master iteration counts of the decomposition methods. The columns contain average numbers of all iterations, of substantial iterations, and of unsubstantial iterations.

Each of the decomposition methods outperformed the direct solution approach of DEC in our experiments. The effect of regularization seems remarkable. The regularized methods Level-Risk and Level-Risk-ODA proved much faster than the unregularized counterparts Benders-Risk and Benders-Risk-ODA, respectively.

In terms of cumulated running times, the on-demand accuracy approach of Level-Risk-ODA resulted in a slight improvement over the level regularization of Level-Risk. In terms of substantial iteration counts, however, the difference is significant: Level-Risk-ODA performed less than half as many substantial iterations as Level-Risk did. (Second-stage problems are solved only in substantial iterations.) This implies that depending on the size of the second-stage problems, and the solver used for the master problem, the effect of the on-demand approach may become significant.

Concerning unregularized decomposition methods, the on-demand accuracy approach of Benders-Risk-ODA resulted in a 37% reduction in running time over the plain cutting-plane method of Benders-Risk.

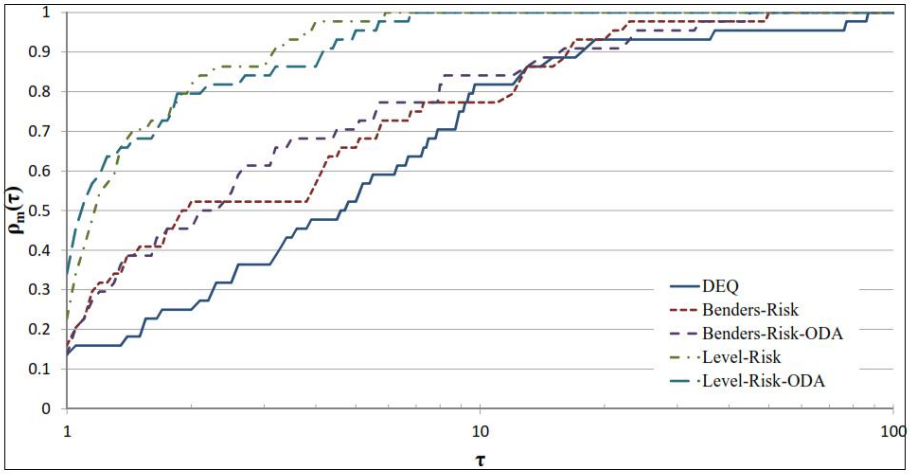

Figure 6.1: Performance profiles of the different methods.

According to the performance profiles shown in Figure 6.1, roughly 35% of the test instances are solved fastest by the regularized on-demand accuracy approach of Level-Risk-ODA, and this method solves about 80% of the instances within twice the time of the fastest method. (The DEQ approach is in more than 50% of the cases at least four times slower than the fastest method.)

Application of the results

We formulated and solved large instances of the CVaR-constrained version of the strategic gas purchase planning problem of [94]. The aim was to hedge against the risk caused by potential low demand. The gas utility company decided not to implement the optimal solution obtained from the risk-averse problems; they preferred an insurance cover. The experiments were still useful, because decision makers could compare the cost of an insurance cover to the decrease in average profit due to a risk constraint.

6.3 Summary

In [64], I generalized the on-demand accuracy approach to risk-averse two-stage problems. I considered two problem types, applying a CVaR constraint or a stochastic ordering constraint, respectively, on the recourse. I reformulated the latter problem using the dominance measure described in Chapter 3.

I adapted the partially inexact version of the constrained level method, recounted in Chapter 2, to the resulting risk-averse problems. The main feature

is that the descent target is a convex combination of the model function value at the new iterate on the one hand, and the best upper estimate known, on the other hand.

In collaboration with Leena Suhl, Achim Koberstein and Christian Wolf from the DS&OR (Decision Support & Operations Research) Lab of Paderborn University, we implemented and compared different methods for the solution of the CVaR-constrained problem (6.1), and performed an extensive computational study.

Each of the decomposition methods outperformed the direct solution approach in our experiments. The effect of regularization proved remarkable. In terms of cumulated running times, the on-demand accuracy approach resulted in a slight improvement over the level regularization. – In terms of substantial iteration counts, however, the difference was significant. (Second-stage problems are solved only in substantial iterations.) This implies that depending on the size of the second-stage problems, and the solver used for the master problem, the effect of the on-demand approach may become significant. – Roughly 35% of the test instances were solved fastest by the regularized on-demand accuracy approach, and this method solved about 80% of the instances within twice the time of the fastest method.

We formulated and solved large instances of the CVaR-constrained version of the real-life strategic gas purchase planning problem of my co-workers. The aim was to hedge against the risk caused by potential low demand (in a mild winter). The gas utility company decided not to implement the optimal solution obtained from the risk-averse problems; they preferred an insurance cover. The experiments were still useful, because decision makers could compare the cost of an insurance cover to the decrease in average profit due to a risk constraint.

Chapter 7

Probabilistic problems

In this chapter I consider probability maximization and probabilistic constrained problems in the respective forms of

$$\max \mathrm{P}\big(\boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{Z}\big) \quad \text{such that} \quad \boldsymbol{x} \in X \tag{7.1}$$

and

$$\min h(\boldsymbol{x}) \quad \text{such that} \quad \boldsymbol{x} \in X,\ \ \mathrm{P}\big(\boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{Z}\big) \geq p, \tag{7.2}$$

where $\boldsymbol{Z}$ denotes an n-dimensional random vector of known distribution, the decision vector is $\boldsymbol{x} \in \mathbb{R}^m$, the feasible domain being $X \subset \mathbb{R}^m$. The functions are $\boldsymbol{g} : \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$, and $p \gg 0$ is a given probability. In many applications, $\boldsymbol{g}(\boldsymbol{x}) = T\boldsymbol{x}$ is a linear function with an appropriate matrix T.

7.1 Historical overview

The probabilistic constrained decision model was introduced by Charnes, Cooper and Symonds [21]. These authors use the term *chance constraint*. Variants and extensions were presented in [20]. Though these early models were based on individual chance constraints. A joint probabilistic constraint based on a random vector having stochastically independent components was considered by Miller and Wagner [109]. The more general problem that allowed stochastically dependent components was introduced by Prékopa [126, 128] and further investigated by him and his followers.

The primary line of research has been to investigate conditions under which the level sets

$$\mathcal{L}_p = \{\, \boldsymbol{z} \mid \mathrm{P}\,(\boldsymbol{z} \geq \boldsymbol{Z}) \geq p \,\} \tag{7.3}$$

or

$$\mathcal{M}_p = \big\{\, \boldsymbol{x} \,\big|\, \mathrm{P}\big(\boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{Z}\big) \geq p \,\big\} \tag{7.4}$$

are convex. Prékopa made a momentous step by developing the theory of log-concave measures [127, 129]. This was later generalized in [12, 14, 168]. A recent advance in this line is the concept eventual convexity [78, 79], where convexity of the level set $\mathcal{M}_p$ is proven under weaker assumptions.

7.2 Solution methods

In [133], Prékopa, Ganczer, Deák and Patyi developed a model (STABIL) for a planning problem in the Hungarian electrical energy sector. The resulting stochastic programming problem was solved by a feasible direction method of Zoutendijk [188]. A non-standard dual formulation for probabilistic constrained problems was proposed by Komáromi [95, 96]. This is a max-min formulation, the inner problem being minimization of a linear function over the level set $\mathcal{L}_p$. For the solution of the dual problem, a special feasible direction method is developed in [95].

Cutting-plane methods were also developed for probabilistic constrained problems, approximating the level set $\mathcal{L}_p$. The method of Prékopa and Szántai [134] applies a Slater point to determine where to construct the next cut. (Namely, the intersection of the boundary of $\mathcal{L}_p$ on the one hand, and the interval connecting the Slater point with the current iterate on the other hand.) The method is related to that of Veinott [178]. In his solver built for the STABIL problem, Szántai [163] developed an interval bisection algorithm for safely computing the intersection point on the boundary of $\mathcal{L}_p$ when probability values cannot be calculated with high precision. He also applied Veinott's technique of moving the Slater point in the course of the solution process, which results in faster convergence and makes the supporting hyperplane method equivalent to a method of Zoutendijk [188]. Mayer [108] proposed a central cutting plane method, an adaptation of Elzinga and Moore [47]. Cutting-plane methods converge in less iterations than feasible direction methods do, since former gradient information is retained.

Prékopa [130] initiated a novel solution approach by introducing the concept of p-efficient points. $\boldsymbol{z}$ is called p-efficient if $F(\boldsymbol{z}) \geq p$ and there exists no $\boldsymbol{z}'$ such that $\boldsymbol{z}' \leq \boldsymbol{z}$, $\boldsymbol{z}' \neq \boldsymbol{z}$, $F(\boldsymbol{z}') \geq p$. Prékopa, Vizvári, and Badics [135] considered problems with random parameters having a discrete finite distribution. They began with enumerating p-efficient points and based on them, built a convex relaxation of the problem.

Dentcheva, Prékopa, and Ruszczyński [40] formulated the probabilistic constraint in a split form: $T\boldsymbol{x} = \boldsymbol{z}$ with $\boldsymbol{z} \in \mathcal{L}_p$; and constructed a Lagrangian dual by relaxing the constraint $T\boldsymbol{x} = \boldsymbol{z}$. The resulting dual functional is the sum of the respective optimal objective values of two simpler problems. The first auxiliary problem is a linear programming problem, and the second one is the minimization of a linear function over the level set $\mathcal{L}_p$. Based on this decomposition, the authors developed a method, called cone generation, that finds new p-efficient points in the course of the optimization process.

As minimization over the level set $\mathcal{L}_p$ entails a substantial computational effort, the master part of the decomposition framework should succeed with as few p-efficient points as possible. Efficient solution methods were developed by Dentcheva, Lai, and Ruszczyński [37] and Dentcheva and Martinez [39]; the latter applies regularization to the master problem. Approximate minimization over the level set $\mathcal{L}_p$ is another enhancement. Dentcheva et al. [37] constructed

approximate p-efficient points through approximating the original distribution by a discrete one. More recently, Van Ackooij et al. [173] employed a special bundle-type method for the solution of the master problem, based on the on-demand accuracy approach of de Oliveira and Sagastizábal [27]. This means working with inexact data and regulating accuracy in the course of the optimization. Approximate p-efficient points with on-demand accuracy were generated employing the integer programming approach of [105].

The approaches mentioned so far exploit convexity of the level sets $\mathcal{L}_p$ or $\mathcal{M}_p$. Though the level sets are often non-convex, convex approximation is sometimes possible, e.g., Pintér [124]. Recent approaches include Nemirovski and Shapiro [113], Ahmed [2].

Deák applied his successive regression approximation method to the solution of probabilistic constrained problems in [31].

Different types of sampling methods also proved useful recently: the approach of uncertain convex programs by Calafiore, Campi, Garatti and Caré [17, 19, 18]; sample average approximation and integer programming by Ahmed, Luedtke, Nemhauser and associates [104, 105, 103].

7.3 Estimating distribution function values and gradients

Most solution methods need an oracle that computes or estimates distribution function values and gradients. Variance reduction Monte Carlo simulation algorithms have originally been developed to be used in feasible direction and outer cutting plane methods for probabilistic constrained problems. An abundant stream of research in this direction has been initiated by the models, methods and applications pioneered by Prékopa and his school. Normal distribution played a focal role.

Deák's method. This method was published in [29], [30]. Its main thrust is to decompose the normal random vector into two parts, a direction and a distance from the origin. This decomposition can be used both in the generation of sample points and in the calculation of the probability content of a rectangle. It is known that the direction is uniformly distributed on the n-dimensional unit sphere, and the distance from the origin has a chi-distribution with n degrees of freedom. The two parts are independent of each other. The advantage of this method is that it computes the probability content of the rectangle in a 'line section to line section' way, instead of a 'point to point' way.

Szántai's method. The method was published in [161], [162], [163]. It is quoted in Sections 6.5 and 6.6 of Prékopa's book [132]. This procedure can be applied to any multivariate probability distribution function. The only condition is that we have to be able to calculate the one- and the two-dimensional marginal

probability distribution function values. Accuracy can easily be controlled by changing the sample size.

As we have

$$F(z_1,\ldots,z_n) = \mathrm{P}\left(Z_1 \le z_1,\ldots,Z_n \le z_n\right) = 1 - \mathrm{P}\left(\overline{A}_1 \cup \cdots \cup \overline{A}_n\right),$$

where $\overline{A}_i = \{Z_i < z_i\}$ $(i = 1,\ldots,n)$, we can apply bounding and simulation results for the probability of the union of the events. If μ denotes the number of those events which occur out of the events $\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_n$, then the random variable

$$\nu_0 = \begin{cases} 0, & \text{if } \mu = 0 \\ 1, & \text{if } \mu \ge 1 \end{cases}$$

obviously has expected value $\overline{P} = \mathrm{P}(\overline{A}_1 \cup \overline{A}_2 \cup \cdots \cup \overline{A}_n)$.

Further two random variables having expected value $\overline{P}$ can be defined by taking the differences between the true probability value and its second order lower and upper Boole–Bonferroni bounds. The definitions of these bounds can be found in Chapter 6 of [132].

We can estimate the expected value of these three random variables in the same Monte Carlo simulation procedure and so we get three different estimates for the probability value $\overline{P}$. If we estimate the pairwise covariances of these estimates it will be easy to get a final, minimal variance estimate, too. The technique of this is well known in the simulation literature, it is called regression technique.

Gassmann [68] combined Szántai's general algorithm and Deák's algorithm into a hybrid algorithm. The efficiency of this algorithm was explored in [34].

One can use higher order Boole-Bonferroni bounds, too. It will further reduce the variance of the final estimation. However, the necessary CPU time increases, which may reduce the overall efficiency of the resulting estimation. Many new bounds for the probability of the union of events has been developed in the last two decades. These bounds use not only the aggregated information of the first few binomial moments but they also use the individual product event probabilities which sum up the binomial moments. The most important results of this type can be found in the papers by [83], [186], [169], [136], [15], [16], [13] and [107]. Szántai in [165] showed that the efficiency of his variance reduction technique can be improved significantly if one uses some of the above listed bounds.

Genz's method. Genz in [70] deals with the estimation of the multivariate normal probability content of a rectangle, a problem more general than the calculation of multivariate probability distribution function values. The main idea is to transform the integration region to the unit cube $[0,1]^n$ by a sequence of elementary transformations. Genz describes three different methods for solving this transformed integral. The first method is based on a polynomial approximation of the integrand. For better performance, the unit cube is split into subregions which are subsequently partitioned further whenever the approximation is not accurate enough. The second method uses quasi-random

integration points. Finally, the third method uses pseudo-random integration points which results in error estimates being statistical in nature.

Gradient computation. If a multivariate probability distribution function is differentiable everywhere then its partial derivatives have the general formula

$$\frac{\partial F(z_1,\ldots,z_n)}{\partial z_i} = F(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n|\ z_i)f_i(z_i), \tag{7.5}$$

where $F(z_1,\ldots,z_n)$ is the probability distribution function of the random variables $\xi_1,\ldots,\xi_n$, and $f_i(z)$ is the probability density function of the random variable ξ_i. $F(z_1,\ldots,\ z_{i-1},z_{i+1},\ldots,z_n|\ z_i)$ is the conditional probability distribution function of the random variables $\xi_1,\ldots,\xi_{i-1}$, $\xi_{i+1},\ldots,\xi_n$, given that $\xi_i = z_i$. This is a well-known formula, see, e.g., section 6.6.4 in Prékopa's book [132].

It is known that any conditional probability distribution of the multivariate normal probability distribution is also normal. Hence we can compute gradient components using (7.5).

7.4 Contribution

In [55], I proposed a polyhedral approximation of the epigraph of the probabilistic function. This approach is analogous to the use of p-efficient points (has actually been motivated by that concept). The dual function is constructed and decomposed in the manner of Dentcheva, Prékopa and Ruszczyński [40], but the nonlinear subproblem is easier. In [40], finding a new p-efficient point amounts to minimization over the level set $\mathcal{L}_p$. In contrast, a new approximation point in [55] is found by unconstrained minimization. Moreover, a practical approximation scheme was developed in the latter paper: instead of exactly solving an unconstrained subproblem occurring during the process, just a single line search proved sufficient. The approach is easy to implement and endures noise in gradient computation.

My coauthors were Edit Csizmás, Rajmund Drenyovszki, Wim van Ackooij, Tibor Vajnai, Lóránt Kovács and Tamás Szántai. Wim van Ackooij works for EDF Research and Development, France. Tamás Szántai is professor emeritus at the Budapest University of Technology and Economics. The rest of the coauthors are my colleagues at the John von Neumann University. The model problems and optimization methods were developed by me. Wim van Ackooij collaborated in compiling a historical overview, and in test problem selection. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues concerning the oracle and finding a starting solution.

The paper [55] deals with an n-dimensional nondegenerate normal probability distribution. Let $F(\boldsymbol{z})$ denote the distribution function. Due to logconcavity

of the normal distribution, the probabilistic function $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ is convex. We discuss a probability maximization problem in the form

$$\min\ \phi(T\boldsymbol{x}) \quad \text{subject to} \quad A\boldsymbol{x} \leq \boldsymbol{b}, \tag{7.6}$$

where vectors are $\boldsymbol{x} \in \mathbb{R}^m,\ \boldsymbol{b} \in \mathbb{R}^r$, and the matrices T and A are of sizes $n \times m$ and $r \times m$, respectively. We assume that the feasible domain is not empty and is bounded.

Exploiting the monotonicity of the objective function, problem (7.6) can be written as

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \ \boldsymbol{z} - T\boldsymbol{x} \leq \boldsymbol{0}. \tag{7.7}$$

This problem has an optimal solution because the feasible domain of (7.6) is nonempty and bounded. Introducing the multiplier vector $-\boldsymbol{y} \in \mathbb{R}^r, -\boldsymbol{y} \geq \boldsymbol{0}$ to the constraint $A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0}$, and $-\boldsymbol{u} \in \mathbb{R}^n, -\boldsymbol{u} \geq \boldsymbol{0}$ to the constraint $\boldsymbol{z} - T\boldsymbol{x} \leq \boldsymbol{0}$, the Lagrangian dual of (7.7) can be written as

$$\max\ \{\boldsymbol{y}^T\boldsymbol{b} - \phi^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}, \tag{7.8}$$

where

$$\mathcal{D} := \left\{\ (\boldsymbol{y}, \boldsymbol{u}) \in \mathbb{R}^{r+n}\ \mid\ \boldsymbol{y}, \boldsymbol{u} \leq \boldsymbol{0},\ \ T^T\boldsymbol{u} = A^T\boldsymbol{y}\ \right\}. \tag{7.9}$$

According to the theory of convex duality, this problem has an optimal solution.

Polyhedral models

Suppose we have evaluated the function $\phi(\boldsymbol{z})$ at points $\boldsymbol{z}_i$ $(i = 0, 1, \ldots, k)$; we introduce the notation $\phi_i = \phi(\boldsymbol{z}_i)$ for respective objective values. An inner approximation of $\phi(.)$ is

$$\begin{aligned} \phi_k(\boldsymbol{z}) = \quad & \min\ \sum_{i=0}^{k} \lambda_i \phi_i \\ & \text{such that} \\ & \lambda_i \geq 0\ (i = 0, \ldots, k),\ \ \sum_{i=0}^{k} \lambda_i = 1, \quad \sum_{i=0}^{k} \lambda_i \boldsymbol{z}_i = \boldsymbol{z}. \end{aligned} \tag{7.10}$$

If $\boldsymbol{z} \notin \mathrm{Conv}(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k)$, then let $\phi_k(\boldsymbol{z}) := +\infty$. A polyhedral model of problem (7.7) is

$$\min\ \phi_k(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \boldsymbol{z} - T\boldsymbol{x} \leq \boldsymbol{0}. \tag{7.11}$$

We assume that (7.11) is feasible, i.e., its optimum is finite. This can be ensured by proper selection of the initial $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k$ points. The convex conjugate of $\phi_k(\boldsymbol{z})$ is

$$\phi_k^\star(\boldsymbol{u}) = \max_{0 \leq i \leq k} \{\boldsymbol{u}^T\boldsymbol{z}_i - \phi_i\}. \tag{7.12}$$

As $\phi_k^\star(.)$ is a cutting-plane model of $\phi^\star(.)$, the following problem is a polyhedral model of problem (7.8):

$$\max\ \{\boldsymbol{y}^T\boldsymbol{b} - \phi_k^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}. \tag{7.13}$$

Linear programming formulations

The primal model problem (7.10)-(7.11) will be formulated as

$$\begin{array}{lll} \min & \sum_{i=0}^{k} \phi_i \lambda_i & \\ \text{such that} & \lambda_i \geq 0 & (i = 0, \ldots, k), \\ & \sum_{i=0}^{k} \lambda_i & = 1, \\ & \sum_{i=0}^{k} \lambda_i \boldsymbol{z}_i \quad -T\boldsymbol{x} & \leq \boldsymbol{0}, \\ & A\boldsymbol{x} & \leq \boldsymbol{b}. \end{array} \tag{7.14}$$

The dual model problem (7.12)-(7.13), formulated as a linear programming problem, is just the LP dual of (7.14):

$$\begin{array}{lllll} \max & \vartheta & + \boldsymbol{b}^T \boldsymbol{y} & & \\ \text{such that} & \boldsymbol{u}, & \boldsymbol{y} & \leq \boldsymbol{0}, & \\ & \vartheta \quad + \boldsymbol{z}_i^T \boldsymbol{u} & & \leq \phi_i & (i = 0, \ldots, k), \\ & - T^T \boldsymbol{u} & + A^T \boldsymbol{y} & = \boldsymbol{0}. & \end{array} \tag{7.15}$$

Let $(\overline{\lambda}_0, \ldots, \overline{\lambda}_k, \overline{\boldsymbol{x}})$ and $(\overline{\vartheta}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$ denote respective optimal solutions of the problems (7.14) and (7.15) – both existing due to our assumption concerning the feasibility of (7.11) and hence (7.14). Let moreover

$$\overline{\boldsymbol{z}} = \sum_{i=0}^{k} \overline{\lambda}_i \boldsymbol{z}_i. \tag{7.16}$$

Observation 27 *We have*

(a) $\phi_k(\overline{\boldsymbol{z}}) = \sum_{i=0}^{k} \phi_i \overline{\lambda}_i = \overline{\vartheta} + \overline{\boldsymbol{u}}^T \overline{\boldsymbol{z}}$,

(b) $\overline{\vartheta} = -\phi_k^\star(\overline{\boldsymbol{u}})$,

(c) $\phi_k(\overline{\boldsymbol{z}}) + \phi_k^\star(\overline{\boldsymbol{u}}) = \overline{\boldsymbol{u}}^T \overline{\boldsymbol{z}}$ *and hence* $\overline{\boldsymbol{u}} \in \partial \phi_k(\overline{\boldsymbol{z}})$.

Proof.

(a) The first equality follows from the equivalence of problems (7.14) and (7.11). The second equality is a straight consequence of complementarity. Indeed,

$\overline{\lambda}_i > 0$ implies that the corresponding reduced cost component is 0 in (7.14), i.e., $\overline{\vartheta} + \overline{\boldsymbol{u}}^T \boldsymbol{z}_i = \phi_i$. It follows that

$$\sum_{i=0}^{k} \phi_i \overline{\lambda}_i \;=\; \sum_{i=0}^{k} \left(\overline{\vartheta} + \overline{\boldsymbol{u}}^T \boldsymbol{z}_i\right) \overline{\lambda}_i \;=\; \overline{\vartheta} \sum_{i=0}^{k} \overline{\lambda}_i \;+\; \overline{\boldsymbol{u}}^T \sum_{i=0}^{k} \overline{\lambda}_i \boldsymbol{z}_i.$$

(b) follows from the equivalence of problems (7.15) and (7.13).

(c) The equality is a consequence of *(a)* and *(b)*. This is Fenchel's equality between $\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{z}}$, with respect to the model function $\phi_k(.)$. The statement on $\overline{\boldsymbol{u}}$ being a subgradient is part of Theorem 23.5 in Rockafellar's book [140].

A column generation procedure

An optimal dual solution (i.e., shadow price vector) of the current model problem is $(\overline{\vartheta}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$. Given a vector $\boldsymbol{z} \in \mathbb{R}^n$, we can add a new column in (7.14), corresponding to $\boldsymbol{z}_{k+1} = \boldsymbol{z}$. This is an improving column if its reduced cost

$$\overline{\rho}(\boldsymbol{z}) \;:=\; \overline{\vartheta} \;+\; \overline{\boldsymbol{u}}^T \boldsymbol{z} - \phi(\boldsymbol{z}) \tag{7.17}$$

is positive. – It is easily seen that the reduced cost of $\overline{\boldsymbol{z}}$ is non-negative. Indeed,

$$\overline{\rho}(\overline{\boldsymbol{z}}) \;\geq\; \overline{\vartheta} \;+\; \overline{\boldsymbol{u}}^T \overline{\boldsymbol{z}} - \phi_k(\overline{\boldsymbol{z}}) \;= 0 \tag{7.18}$$

follows from $\phi_k(.) \geq \phi(.)$ and Observation 27 *(a)*.

In the context of the simplex method, the Markowitz column-selection rule is widely used. The Markowitz rule selects the vector with the largest reduced cost. Coming back to the present problem (7.14), let

$$\overline{\mathcal{R}} := \max_{\boldsymbol{z}} \overline{\rho}(\boldsymbol{z}). \tag{7.19}$$

The column with the largest reduced cost can, in principle, be found by a steepest descent method applied to the function $-\overline{\rho}(\boldsymbol{z})$.

Remark 28 *Looking at the column generation approach from a dual viewpoint, we can see a cutting-plane method applied to the convex dual problem (7.8). The convex conjugate function $\phi^\star(\boldsymbol{u})$ is approximated with the polyhedral model function $\phi_k^\star(\boldsymbol{u})$. From this dual viewpoint, the maximization problem (7.19) is interpreted as finding the cut that, at the current iterate $\overline{\boldsymbol{u}}$, cuts deepest into the epigraph of $\phi_k^\star(\boldsymbol{u})$. – This relationship between the primal and dual approaches is well known, see, e.g., [65, 66].*

Numerical considerations

If $\boldsymbol{z}$ is such that $F(\boldsymbol{z})$ is near zero, then the computation of the objective value $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ is problematic because the logarithm amplifies any errors in $F(\boldsymbol{z})$. Moreover, as we have $\nabla\phi(\boldsymbol{z}) = -\frac{1}{F(\boldsymbol{x})} \nabla F(\boldsymbol{x})$, gradient computation is also error-prone for such $\boldsymbol{z}$. To avoid these difficulties, we work under

Assumption 29 *A significantly high probability can be achieved in the probability maximization problem. Specifically, a feasible point $\check{\boldsymbol{z}}$ is known such that $F(\check{\boldsymbol{z}}) \geq 0.5$.*

By including $\check{\boldsymbol{z}}$ of Assumption 29 among the initial columns of the master problem, we always have $F(\overline{\boldsymbol{z}}) \geq 0.5$ with the current solution $\overline{\boldsymbol{z}}$ defined in (7.16). Hence $\phi(\overline{\boldsymbol{z}})$ can be computed with a high accuracy.

We perform a single line search in each column generation subproblem, starting always from the current $\overline{\boldsymbol{z}}$. It means that a high-quality estimate can be generated for the gradient, which designates the direction of the line search. Once the direction of the search is determined, we only work with function values (there is no need for any further gradient information in the current column generation subproblem). The line search is performed with a high accuracy over the region $\mathcal{L}(F, 0.5) = \{\, \boldsymbol{z} \,|\, F(\boldsymbol{z}) \geq 0.5 \,\}$ which includes the optimal solution of the probability maximization problem (7.7).

We can carry on with the line search even if we have left the safe region $\mathcal{L}(F, 0.5)$. Given a point $\hat{\boldsymbol{z}}$ along the search ray, let $\hat{p} > 0$ be such that $\hat{p} \leq F(\hat{\boldsymbol{z}})$ holds almost surely. (Simulation procedures generally provide a confidence interval together with an estimate.) If the vector $\hat{\boldsymbol{z}}$ is to be included in the master problem (7.14) as a new column, then we set the corresponding cost coefficient as $\phi = -\log \hat{p}$. Under such an arrangement, our model remains consistent, i.e., the model function $\phi_k(\boldsymbol{z})$ is almost surely an inner approximation of the probabilistic function $\phi(\boldsymbol{z})$.

Computational experiments and a heuristic improvement

We implemented the column generation procedure in MATLAB. The master problem was solved with the IBM ILOG CPLEX (version 12.6.3) optimization toolbox. The column generation subproblems (7.19) were solved by a steepest descent method applied to the probabilistic function $\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}^T \boldsymbol{z}$, starting from $\overline{\boldsymbol{z}}$ of (7.16). Multivariate normal distribution function values were computed by the QSIMVNV MATLAB function implemented by Genz [70]. Gradients were computed componentwise, according to the formula (7.5).

We tested the method on ten problems, each having normally distributed random parameters. We derived 9 test problems from the coffee blending model of Szántai [164]. In these problems, the random parameters had up to 5 components. Moreover we tested the method on a cash matching problem with fifteen dimensional normal probability distribution. (In this problem we are interested in investing a certain amount of cash on behalf of a pension fund that needs to make certain payments over the coming 15 years of time.) Details of the cash matching problem can be found in [37] and [77]. – All these problems had originally been formulated as probabilistic constrained cost minimization problems, but we transformed them into probability maximization problems by introducing cost constraints.

Probabilistic function values were computed with high accuracy in our computational experiments. (We had $F(\boldsymbol{z}_i) \geq 0.9$ with each column $\boldsymbol{z}_i$ generated

in the column generation scheme.) Our simple implementation reliably solved the test problems. Iteration counts were comparable to problem dimensions. But we found that most of the computational effort was spent in the Genz subroutine. In order to balance different efforts, we decided to apply rough approximate solutions for the column generation subproblems, instead of the high-precision solutions of the first test round. We performed just a single line search in each column generation subproblem, hence a single gradient computation was performed for each new column. (Even the single line search was approximate.) This way the solution effort of an individual column generation decreased substantially. The application of this heuristic procedure never resulted in any substantial increase in the number of new columns needed to solve a test problem. I found this interesting and investigated possible causes.

Justification of the heuristic improvement

Let us consider an idealized setting, with a well-conditioned function $f : \mathbb{R}^n \to \mathbb{R}$. By well-conditioned, I mean that the following assumption holds.

Assumption 30 *The function $f(\boldsymbol{z})$ is twice continuously differentiable, and real numbers α, ω $(0 < \alpha \leq \omega)$ are known such that*

$$\alpha I \preceq \nabla^2 f(\boldsymbol{z}) \preceq \omega I \qquad (\boldsymbol{z} \in \mathbb{R}^n).$$

Here $\nabla^2 f(\boldsymbol{z})$ is the Hessian matrix, I is the identity matrix, and the relation $U \preceq V$ between matrices means that $V - U$ is positive semidefinite.

We wish to minimize f over $\mathbb{R}^n$. The efficiency of the steepest descent method can be estimated on the basis of the following well-known theorem that can be found e.g., in Chapter 8.6 of [106]. ([151] in Chapter 5.3.5, Theorem 5.7 presents a slightly different form.)

Theorem 31 *Let Assumption 30 hold. We minimize $f(\boldsymbol{z})$ over $\mathbb{R}^n$ using a steepest descent method, starting from a point $\boldsymbol{z}^0$. Let $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^j, \ldots$ denote the iterates obtained by applying exact line search at each step. Then we have*

$$f\left(\boldsymbol{z}^j\right) - \mathcal{F} \leq \left(1 - \frac{\alpha}{\omega}\right)^j \left[\, f\left(\boldsymbol{z}^0\right) - \mathcal{F} \,\right], \tag{7.20}$$

where $\mathcal{F} = \min_{\boldsymbol{z}} f(\boldsymbol{z})$.

In our column generation subproblem, we wish to minimize $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z}) = \phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}^T \boldsymbol{z} - \overline{\vartheta}$. If Assumption 30 should hold for the probabilistic function $\phi(\boldsymbol{z})$, then the objective function of the column generating problem would inherit it. Efficiency of a steepest descent method starting from $\overline{\boldsymbol{z}}$ could then be estimated by Theorem 31. Indeed, substituting $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z})$, $\mathcal{F} = -\overline{\mathcal{R}}$ and $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ in (7.20), and introducing the notation $\varrho = 1 - \alpha/\omega$, we get

$$\overline{\mathcal{R}} - \overline{\rho}\left(\boldsymbol{z}^j\right) \leq \varrho^j \left[\, \overline{\mathcal{R}} - \overline{\rho}\left(\overline{\boldsymbol{z}}\right) \,\right].$$

Due to (7.18), $\overline{\rho}\,(\overline{\boldsymbol{z}})$ can be discarded in the right-hand side, and we get

$$\overline{\rho}\,\big(\boldsymbol{z}^j\big) \;\geq\; \big(1-\varrho^j\big)\,\overline{\mathcal{R}} \qquad \text{for} \quad j=1,2,\ldots,$$

certifying a fast convergence. In view of the Markowitz rule mentioned above, we find a fairly good improving vector in the column generation scheme in a very few iterations, provided the condition number α/ω is not extremely bad. Setting $j=1$ always resulted in a good improving vector in our computational experiments.

Assumption 30 of course does not hold throughout $\mathbb{R}^n$ for the function $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ when $F(\boldsymbol{z})$ is a distribution function. However, the proof of Theorem 31, as related in Chapter 8.6 of [106], actually requires bounded Hessians only in a certain neighbourhood. I conjecture that the present objective function is well-conditioned in an area where potential optimal solutions typically belong. The following example makes a case for this conjecture. (Below, in Remark 56 of Section 10.1, I describe a means of regularizing a poorly conditioned objective.) The column generation procedure gains traction as an optimal solution is gradually approached.

Example 32 *I illustrate the well-conditioned nature of* $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ *in case* $F(\boldsymbol{z})$ *is a two-dimensional standard normal distribution function, where the covariance between the marginals is* 0.5.

The two Figures 7.1 depict the smaller and the larger eigenvalue, respectively, of the Hessian matrix $\nabla^2\phi(\boldsymbol{z})$ *as a function of* $\boldsymbol{z}$. *In the left-hand figure (smaller eigenvalue), contour lines from top right are* $1e-5, 1e-4, 1e-3, 1e-2$. *In the area not shaded gray, the smaller eigenvalue is above* $1e-5$.

In the right-hand figure (larger eigenvalue), contour lines from top right are $1, 1.2, 1.4, 1.6$. *In the area not shaded gray, the larger eigenvalue is below* 1.6.

Figure 7.2 shows contour lines of the two-dimensional normal distribution function $F(\boldsymbol{z})$. *From top right, these contour lines belong to the probabilities* $0.99, 0.95,$ *and* 0.90, *respectively. The critical area for the corresponding p-efficient points is the neighbourhood of* $\boldsymbol{z}=(2,2)$, *where* $\phi(\boldsymbol{z})$ *is well-conditioned.*

7.5 Summary

In [55], I proposed a polyhedral approximation of the epigraph of the probabilistic function. I worked out a successive approximation scheme for probability maximization; new approximation points are added as the process progresses. In case of linear constraints, it is a column generation scheme for a linear programming master problem. (From a dual point of view, the column generation approach is a cutting-plane method applied to a conjugate function.)

This project was motivated by the concept of p-efficient points proposed by Prékopa [130], and the approximation scheme is analogous to the cone generation scheme of Dentcheva, Prékopa and Ruszczyński [40]. But in my scheme,

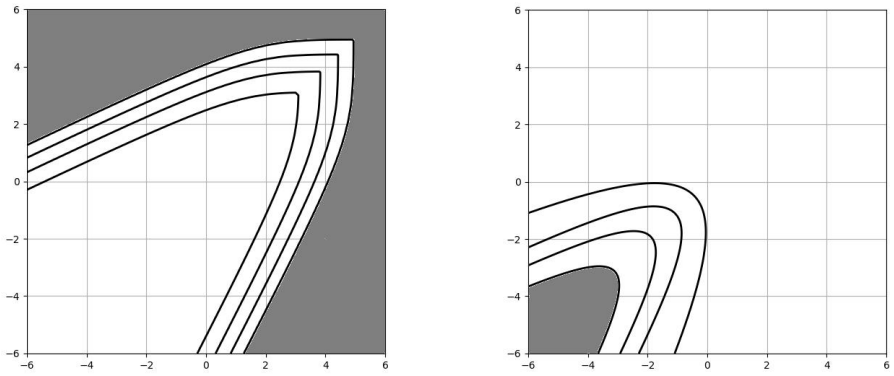

Figure 7.1: Contour lines of the smaller and the larger eigenvalue, respectively, of $\nabla^2[-\log F(\boldsymbol{z})]$ as a function of $\boldsymbol{z}$, for a two-dimensional normal distribution function $F(\boldsymbol{z})$ $(-6 \leq z_1, z_2 \leq +6)$.

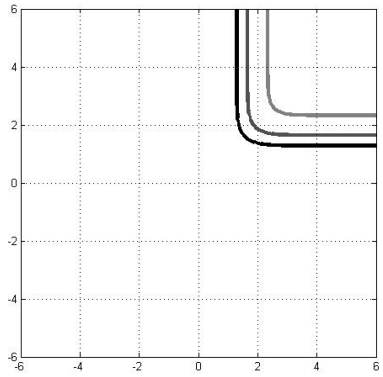

Figure 7.2: Contour lines of the two-dimensional normal distribution function $F(\boldsymbol{z})$.

the subproblem of finding a new approximation point is easier; it is an unconstrained convex optimization problem, solvable by a simple gradient descent method. The approach is easy to implement and endures noise in gradient computation. Hence the classic probability estimation methods are applicable.

My coauthors were Edit Csizmás, Rajmund Drenyovszki, Wim van Ackooij, Tibor Vajnai, Lóránt Kovács and Tamás Szántai. Wim van Ackooij works for EDF Research and Development, France. Tamás Szántai is professor emeritus at the Budapest University of Technology and Economics. The rest of the

coauthors are my colleagues at the John von Neumann University. The model problems and optimization methods were developed by me. Wim van Ackooij collaborated in compiling a historical overview, and in test problem selection. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues concerning the oracle and finding a starting solution.

Most of the computational effort in our tests was spent in gradient computation. In order to balance different efforts, we decided to apply rough approximate solutions for the column generation subproblems, performing just a single line search in each gradient descent method. This heuristic procedure never resulted in any substantial increase in the number of new columns needed to solve a test problem. I found this interesting and investigated possible causes.

The gradient descent method is remarkably effective in case the objective function is well-conditioned in a certain neighbourhood of the optimal solution. I make a case for conjecturing that the probabilistic function is well-conditioned in an area where potential optimal solutions typically belong. (The bounded formulation of Chapter 10 allows the regularization of a poorly conditioned objective.) The column generation procedure gains traction as an optimal solution is gradually approached.

Chapter 8

A randomized method for handling a difficult objective function

This project was motivated by our experiments with the inner approximation approach to probability maximization, related in Chapter 7. I consider a minimization problem in the abstract form (7.6), with a convex objective function $\phi(\boldsymbol{z})$ whose gradient computation is taxing.

In this chapter I propose a randomized version of the column generation scheme of Chapter 7 in an idealized setting, assuming that the objective function has bounded Hessians. I assume moreover that appropriate gradient estimates can be constructed by simulation, with a reasonable effort. – For the sake of simplicity, I assume that objective values are computed with a high precision.

As the proposed method bears an analogy to stochastic gradient methods, I present a brief overview of the latter family. Then I describe the randomized column generation scheme. I also include an error analysis and reliability considerations.

As mentioned above, I assume in this chapter that the objective function $\phi(\boldsymbol{z})$ has bounded Hessians. On the other hand, I do not exploit monotonicity of $\phi(\boldsymbol{z})$. Hence minor modifications are needed on the problems and models (7.7 - 7.15) of Chapter 7. Namely, variable splitting will be formulated as $\boldsymbol{z}-T\boldsymbol{x}=\boldsymbol{0}$, and, consequently, there will be no sign restriction on the dual variable $\boldsymbol{u}$. For convenience, I include the modified problems. Problem (7.7) is written as

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x}-\boldsymbol{b}\leq\boldsymbol{0},\ \ \boldsymbol{z}-T\boldsymbol{x}=\boldsymbol{0}. \tag{8.1}$$

The Lagrangian dual of (8.1) is

$$\max\ \{\boldsymbol{y}^T\boldsymbol{b}-\phi^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y},\boldsymbol{u})\in\mathcal{D}', \tag{8.2}$$

where

$$\mathcal{D}' := \left\{ (\boldsymbol{y}, \boldsymbol{u}) \in \mathbb{R}^{r+n} \mid \boldsymbol{y} \leq \boldsymbol{0},\ T^T\boldsymbol{u} = A^T\boldsymbol{y} \right\}. \tag{8.3}$$

A polyhedral model of problem (8.1) is

$$\min\ \phi_k(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}. \tag{8.4}$$

Like in Chapter 7, we assume that (8.4) is feasible. The dual model problem is

$$\max \{\boldsymbol{y}^T\boldsymbol{b} - \phi_k^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}'. \tag{8.5}$$

The primal model problem (8.4), formulated in linear programming form, is

$$\begin{array}{lll} \min & \sum_{i=0}^{k} \phi_i\lambda_i & \\ \text{such that} & \lambda_i \geq 0 & (i = 0, \ldots, k), \\ & \sum_{i=0}^{k} \lambda_i & = 1, \\ & \sum_{i=0}^{k} \lambda_i\boldsymbol{z}_i \quad -T\boldsymbol{x} & = \boldsymbol{0}, \\ & A\boldsymbol{x} & \leq \boldsymbol{b}. \end{array} \tag{8.6}$$

The dual model problem (8.5), formulated as a linear programming problem, is just the LP dual of (8.6):

$$\begin{array}{lllll} \max & \vartheta & + \boldsymbol{b}^T\boldsymbol{y} & & \\ \text{such that} & & \boldsymbol{y} & \leq \boldsymbol{0}, & \\ & \vartheta \quad + \boldsymbol{z}_i^T\boldsymbol{u} & & \leq \phi_i & (i = 0, \ldots, k), \\ & - T^T\boldsymbol{u} & + A^T\boldsymbol{y} & = \boldsymbol{0}. & \end{array} \tag{8.7}$$

Let $(\overline{\lambda}_0, \ldots, \overline{\lambda}_k, \overline{\boldsymbol{x}})$ and $(\overline{\vartheta}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$ denote respective optimal solutions of the problems (8.6) and (8.7). Let moreover $\overline{\boldsymbol{z}}$ be derived according to (7.16).

With these objects, Observation 27 remains valid, and the column generation procedure of Chapter 7 remains workable.

8.1 The broader context: stochastic gradient methods

The idea of stochastic approximation goes back to Robbins and Monro [139]. Ermoliev has been the major proponent of the method, together with his co-workers. Many relevant articles can be found in the volume [49] edited by Ermoliev and Wets.

The area is under active development ever since. The approach is attractive from a theoretical point of view, but early forms might perform poorly in practice. Recent forms combine theoretical depth with practical effectiveness.

As a recent example of the stochastic gradient approach, I sketch the robust stochastic approximation method of Nemirovski and Yudin [114]. The problem is formulated as

$$\min\ f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} \in X, \tag{8.8}$$

where $X \subset \mathbb{R}^n$ is a convex compact set, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex differentiable function. It is assumed that, given $\boldsymbol{x} \in X$, realizations of a random vector $\boldsymbol{G}$ can be constructed such that $\mathrm{E}(\boldsymbol{G}) = \nabla f(\boldsymbol{x})$, and $\mathrm{E}(\|\boldsymbol{G}\|^2) \leq M^2$ holds with a constant M independent of $\boldsymbol{x}$.

The method is iterative, and a starting point $\boldsymbol{x}_1 \in X$ is needed. Let $\boldsymbol{x}_k \in X$ denote the kth iterate, and $\boldsymbol{G}_k$ a random estimate of the corresponding gradient $\boldsymbol{g}_k = \nabla f(\boldsymbol{z}_k)$. Gradient estimates for different iterates are based on independent, identically distributed samples. The next iterate is computed as

$$\boldsymbol{x}_{k+1} = \Pi_X \left(\boldsymbol{x}_k - h_k \boldsymbol{G}_k \right), \tag{8.9}$$

where $h_k > 0$ is an appropriate step length, and Π_X denotes projection onto X, i.e., $\Pi_X(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}' \in X} \|\boldsymbol{x} - \boldsymbol{x}'\|$.

Nemirovski and Yudin prove different convergence results; from our present point of view, the most relevant one is the following. Suppose that we wish to perform N steps with the above procedure, and set step length to be constant:

$$h_k = \frac{\mathrm{diag}(X)}{M\sqrt{N}}, \tag{8.10}$$

where $\mathrm{diag}(X)$ is the longest (Euclidean) distance occurring in X. Then we have

$$\mathrm{E}\Big(f(\overline{\boldsymbol{x}}_N) \Big) - \mathcal{F} \ \leq \ \frac{M \cdot \mathrm{diag}(X)}{\sqrt{N}}, \tag{8.11}$$

where $\mathcal{F}$ denotes the minimum of (8.8), and

$$\overline{\boldsymbol{x}}_N = \sum_{j=1}^{N} \lambda_j^N \boldsymbol{x}_j \quad \text{with} \quad \lambda_j^N = \frac{h_j}{\sum_{j=1}^{N} h_j}. \tag{8.12}$$

8.2 Contribution: a randomized column generation scheme

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The results recounted in this section are my contribution.

I extend the column generation scheme of Chapter 7 to handle gradient estimates. We solve problem (7.6) in an idealized setting. We assume that the objective function $\phi(\boldsymbol{z})$ has bounded Hessians, and that we can construct unbiased gradient estimates having bounded variance.

Specifically, we need to approximately solve the column generation subproblem (7.19), i.e., to find a reliable near maximizer of the function $\overline{\rho}(\boldsymbol{z}) = \overline{\vartheta} + \overline{\boldsymbol{u}}^T\boldsymbol{z} - \phi(\boldsymbol{z})$. We apply a stochastic descent method to $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z})$.

$f(\boldsymbol{z})$ inherits bounded Hessians from $\phi(\boldsymbol{z})$, hence Assumption 30 holds. Gradients of $f(\boldsymbol{z})$ have the form $\nabla\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}$. The further the column generation procedure progresses, the smaller the gradient norm gets. To satisfy the requirement of bounded variance, better and better estimates are needed. Hence the assumption on the construction of unbiased gradient estimates having bounded variance is formulated as

Assumption 33 *Given $\boldsymbol{z}, \boldsymbol{u} \in \mathbb{R}^n$, the function value $\phi(\boldsymbol{z})$ can be computed with a high precision (exactly for practical purposes), and the norm $\|\nabla\phi(\boldsymbol{z}) - \boldsymbol{u}\|$ can be computed with a pre-defined relative accuracy. Moreover, realizations of an unbiased stochastic estimate $\boldsymbol{G}$ of the gradient vector $\nabla\phi(\boldsymbol{z})$ can be constructed such that $E\left(\|\boldsymbol{G} - \nabla\phi(\boldsymbol{z})\|^2\right)$ remains below a pre-defined tolerance. (Higher accuracy in case of norm estimation, and tighter tolerance on variance entail larger computational effort.)*

The above assumption specializes to $f(\boldsymbol{z}) = \phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}^T\boldsymbol{z} - \overline{\vartheta}$ as

Assumption 34 *Let $\boldsymbol{z}^\circ \in \mathbb{R}^n$ denote an iterate, and $\boldsymbol{g}^\circ = \nabla f(\boldsymbol{z}^\circ)$ the corresponding gradient. Given $\sigma > 0$, we can construct realizations of a random vector $\boldsymbol{G}^\circ$, satisfying*

$$E(\boldsymbol{G}^\circ) = \boldsymbol{g}^\circ \qquad \text{and} \qquad E\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) \;\leq\; \sigma^2\,\|\boldsymbol{g}^\circ\|^2\,. \tag{8.13}$$

From (8.13) follows

$$\mathrm{E}\left(\|\boldsymbol{G}^\circ\|^2\right) \;=\; \mathrm{E}\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) + \|\boldsymbol{g}^\circ\|^2 \;\leq\; (\sigma^2+1)\,\|\boldsymbol{g}^\circ\|^2\,. \tag{8.14}$$

Theorem 35 *Let Assumptions 30 and 34 hold. We minimize $f(\boldsymbol{z})$ over $\mathbb{R}^n$. We apply a steepest descent method with gradient estimates: at the current iterate $\boldsymbol{z}^\circ$, a gradient estimate $\boldsymbol{G}^\circ$ is generated and a line search is performed in the opposite direction. We assume that gradient estimates at the respective iterates are generated independently.*

Having started from the point $\boldsymbol{z}^0$, and having performed j line searches, let $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^j$ denote the respective iterates. Then we have

$$E\left[f\left(\boldsymbol{z}^j\right)\right] - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega(\sigma^2+1)}\right)^j \left(f\left(\boldsymbol{z}^0\right) - \mathcal{F}\right), \tag{8.15}$$

where $\mathcal{F} = \min_{\boldsymbol{z}} f(\boldsymbol{z})$.

Proof. Let $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^{j-1}$ denote the respective gradient estimates for the iterates $\boldsymbol{z}^0, \ldots, \boldsymbol{z}^{j-1}$.

To begin with, we focus on the first line search whose starting point is $\boldsymbol{z}^\circ = \boldsymbol{z}^0$. Here $\boldsymbol{z}^\circ$ is a given (not random) vector. I adapt the proof of Theorem

31, presented in Chapter 8.6 of [106], to employ the gradient estimate $\boldsymbol{G}^\circ$ instead of the gradient $\boldsymbol{g}^\circ$. From $\nabla^2 f(\boldsymbol{z}) \preceq \omega I$, it follows that

$$f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right) \;\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; t\,\boldsymbol{g}^{\circ\,T}\boldsymbol{G}^\circ \;+\; \frac{\omega}{2}t^2\,\boldsymbol{G}^{\circ\,T}\boldsymbol{G}^\circ$$

holds for any $t \in \mathbb{R}$ (a consequence of Taylor's theorem). Considering expectations on both sides, we get

$$\begin{aligned}\mathrm{E}\left[f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)\right] \;&\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; t\,\left\|\boldsymbol{g}^\circ\right\|^2 \;+\; \tfrac{\omega}{2}t^2\,\mathrm{E}\left(\left\|\boldsymbol{G}^\circ\right\|^2\right)\\ &\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; t\,\left\|\boldsymbol{g}^\circ\right\|^2 \;+\; \tfrac{\omega}{2}t^2\left(\sigma^2+1\right)\,\left\|\boldsymbol{g}^\circ\right\|^2\end{aligned}$$

according to (8.14). We consider the respective minima in t separately of the two sides. The right-hand side is a quadratic expression, yielding minimum at $t = \frac{1}{\omega(\sigma^2+1)}$. Inequality is inherited to minima, hence

$$\min_t \mathrm{E}\left[f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)\right] \;\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; \frac{1}{2\omega(\sigma^2+1)}\,\left\|\boldsymbol{g}^\circ\right\|^2. \tag{8.16}$$

For the left-hand side, we obviously have

$$\mathrm{E}\left[\,\min_t f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)\right] \;\leq\; \min_t \mathrm{E}\left[f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)\right]. \tag{8.17}$$

(This is analogous to the basic inequality comparing the wait-and-see and the here-and-now approaches for classic two-stage stochastic programing problems, see, e.g., Chapter 4.3 of [10].)

Let $\boldsymbol{z}'$ denote the minimizer of the line search on the left-hand side of (8.17), i.e., $f\left(\boldsymbol{z}'\right) = \min_t f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)$. (Of course $\boldsymbol{z}'$ is a random vector since it depends on $\boldsymbol{G}^\circ$.) Substituting this in (8.17) and comparing with (8.16), we get

$$\mathrm{E}\left[f\left(\boldsymbol{z}'\right)\right] \;\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; \frac{1}{2\omega(\sigma^2+1)}\,\left\|\boldsymbol{g}^\circ\right\|^2.$$

Subtracting $\mathcal{F}$ from both sides results in

$$\mathrm{E}\left[f\left(\boldsymbol{z}'\right)\right] - \mathcal{F} \;\leq\; f\left(\boldsymbol{z}^\circ\right) - \mathcal{F} \;-\; \frac{1}{2\omega(\sigma^2+1)}\,\left\|\boldsymbol{g}^\circ\right\|^2. \tag{8.18}$$

Coming to the lower bound, a well-known consequence of $\alpha I \preceq \nabla^2 f(\boldsymbol{z})$ is

$$\left\|\boldsymbol{g}^\circ\right\|^2 \;\geq\; 2\alpha\,\left(f\left(\boldsymbol{z}^\circ\right) \;-\mathcal{F}\right) \tag{8.19}$$

(see Chapter 8.6 of [106]). Combining this with (8.18), we get

$$\begin{aligned}\mathrm{E}\left[f\left(\boldsymbol{z}'\right)\right] - \mathcal{F} \;&\leq\; f\left(\boldsymbol{z}^\circ\right) - \mathcal{F} \;-\; \tfrac{\alpha}{\omega(\sigma^2+1)}\,\left(f\left(\boldsymbol{z}^\circ\right) \;-\mathcal{F}\right)\\ &=\; \left(1 - \tfrac{\alpha}{\omega(\sigma^2+1)}\right)\left(f\left(\boldsymbol{z}^\circ\right) \;-\mathcal{F}\right).\end{aligned} \tag{8.20}$$

As we have assumed that $\boldsymbol{z}^\circ$ is a given (not random) vector, the right-hand side of (8.20) is deterministic, and the expectation on the left-hand side is considered according to the distribution of $\boldsymbol{G}^\circ$.

Now, let us examine the $(l+1)$th line search (for $1 \leq l \leq j-1$) where the starting point is $\boldsymbol{z}^\circ = \boldsymbol{z}^l$ and the minimizer is $\boldsymbol{z}' = \boldsymbol{z}^{l+1}$. Of course (8.20) holds with these objects also, but now both sides are random variables, depending on the vectors $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^{l-1}$. (The expectation on the left-hand side is a conditional expectation.) We consider the respective expectations of the two sides, according to the joint distribution of $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^{l-1}$. As the random gradient vectors were generated independently, we get

$$\mathrm{E}\left[f\left(\boldsymbol{z}^{l+1}\right)\right] - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega(\sigma^2+1)}\right)\left(\mathrm{E}\left[f\left(\boldsymbol{z}^{l}\right)\right] - \mathcal{F}\right), \tag{8.21}$$

where the left-hand expectation is now taken according to the joint distribution of $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^l$. – This technique of proof is well known in the context of stochastic gradient schemes, see, e.g., [115].

Finally, (8.15) follows from the iterative application of (8.21). □

Remark 36 *In the present idealistic setting of known ω, we could set the next iterate simply as*

$$\boldsymbol{z}' = \boldsymbol{z}^\circ - \frac{1}{\omega(\sigma^2+1)}\boldsymbol{G}^\circ, \tag{8.22}$$

instead of performing a line search along the ray $\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ$, in the proof of Theorem 35. Besides efficiency considerations, an advantage of the line search

$$f\left(\boldsymbol{z}'\right) = \min_t f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right) \tag{8.23}$$

is that the result remains applicable if the actual bound ω is not known.

Application in the column generation scheme

I examine the utility of applying a stochastic descent method to the column generation subproblem (7.19). Gradient estimates at the respective iterates are generated independently. We apply Theorem 35 to $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z})$.

Corollary 37 *Let a tolerance β $(0 < \beta \ll 1)$ and a probability p $(0 < p \ll 1)$ be given. In $O\left(-\log(\beta\, p)\right)$ steps with the stochastic descent method, we find a vector $\widehat{\boldsymbol{z}}$ such that*

$$P\Big(\;\overline{\rho}\left(\widehat{\boldsymbol{z}}\;\right) \;\geq\; (1-\beta)\overline{\mathcal{R}}\Big) \;\geq\; 1-p.$$

Proof. Let $\varrho = 1 - \frac{\alpha}{\omega(\sigma^2+1)}$ with some $\sigma > 0$. Substituting $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ in (8.15) and taking into account (7.18), we get

$$\mathrm{E}\left[\overline{\rho}(\boldsymbol{z}^j)\right] \;\geq\; \left(1 - \varrho^j\right)\overline{\mathcal{R}}.$$

The gap $\overline{\mathcal{R}}$ is obviously non-negative. In case $\overline{\mathcal{R}} = 0$, the starting iterate $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ of the steepest descent method was already optimal, due to (7.18). In what follows we assume $\overline{\mathcal{R}} > 0$. A trivial transformation yields

$$\mathrm{E}\left[1 - \frac{\overline{\rho}(\boldsymbol{z}^j)}{\overline{\mathcal{R}}}\right] \;\leq\; \varrho^j.$$

By Markov's inequality, we get

$$\mathrm{P}\left(\; 1 - \frac{\overline{\rho}(\boldsymbol{z}^j)}{\overline{\mathcal{R}}} \geq \beta\right) \;\leq\; \frac{\varrho^j}{\beta},$$

and a trivial transformation yields

$$\mathrm{P}\Big(\; \overline{\rho}(\boldsymbol{z}^j\;) \;\leq\; (1-\beta)\overline{\mathcal{R}}\Big) \;\leq\; \frac{1}{\beta}\,\varrho^j.$$

Hence

$$\mathrm{P}\Big(\; \overline{\rho}(\boldsymbol{z}^j\;) \;>\; (1-\beta)\overline{\mathcal{R}}\Big) \;\geq\; 1 - \frac{1}{\beta}\,\varrho^j.$$

Performing j steps with j such that $\varrho^j \leq \beta p$ yields an appropriate $\widehat{\boldsymbol{z}} = \boldsymbol{z}^j$. □

Bounding the optimality gap and reliability considerations

When solving a linear programming problem with the simplex method, one usually applies an optimality tolerance on the reduced cost components. For the master problem of the column generation scheme of Chapter 7, this is not just a heuristic rule:

Observation 38 *$\overline{\mathcal{R}}$ of (7.19) is an upper bound on the gap between the respective optima of the model problem (8.6) and the original convex problem (8.1).*

Proof. We have

$$\overline{\mathcal{R}} \;=\; \max_{\boldsymbol{z}}\, \overline{\rho}(\boldsymbol{z}) \;=\; \phi^\star(\overline{\boldsymbol{u}}) - \phi_k^\star(\overline{\boldsymbol{u}}). \tag{8.24}$$

(The second equality follows from the definition of the conjugate function.)

Since $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{y}})$ is a feasible solution of the dual problem (8.2), it follows that (8.24) is an upper bound on the gap between the respective optima of the dual model problem (8.5) and the dual problem (8.2). The observation follows from convex duality. □

Let

$$\overline{\mathcal{B}} := \frac{1}{1-\beta}\overline{\rho}\,(\widehat{\boldsymbol{z}})\,, \tag{8.25}$$

with the β and $\widehat{\boldsymbol{z}}$ of Corollary 37. Concerning the gap between the respective optima of the model problem (8.6) and the original convex problem (8.1), the reliability

$$\mathrm{P}\left(\overline{\mathcal{B}} \;\geq\; \text{'gap'}\right) \tag{8.26}$$

is at least $1 - p$ with the p of Corollary 37.

Assume that our initial model included the columns $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_\iota$. In the course of the column generation scheme, we select further columns according to Corollary 37, with gradient estimates generated independently. Let the parameters σ and β be fixed for the whole scheme, e.g., set $\beta = 0.5$. On the other hand, we keep increasing the reliability of the individual steps during the process, i.e., let $p = p_\kappa$ $(\kappa = \iota + 1, \iota + 2, \ldots)$ decrease with κ.

Example 39 *Let $p_\kappa = (\kappa - \iota + 9)^{-2}$, then we have $\prod_{\kappa=\iota+1}^{\infty}(1 - p_\kappa) = 0.9$. (This is easily proven. I learned it from Szász [166], Vol. II., Chap. X., § 642.)*

To achieve reliability $1 - p_\kappa$ set in Example 39, we need to make $O(\log \kappa)$ steps with the stochastic descent method when selecting the column $\boldsymbol{z}_\kappa$.

We terminate the column generation process when $\overline{\mathcal{B}}$ of (8.25) gets below the prescribed accuracy. With the setting of Example 39, the terminal bound is correct with a probability at least 0.9, regardless of the number of new columns generated over the course of the procedure.

Comparison with stochastic gradient methods

Our present Assumption 30 is much stronger than mere differentiability, hence the convergence estimate of Theorem 35 is naturally stronger than (8.11).

We proved Theorem 35 for unconstrained minimization (over $\mathbb{R}^n$). In our approach, the constraint $A\boldsymbol{x} \leq \boldsymbol{b}$ in the convex problem (7.6) was taken into account through a column generation scheme. Comparing the column generation scheme with the above stochastic gradient approach, a solution of the linear programming model problem (8.6) is analogous to the iterate averaging (8.12) and the projection in (8.9). The analogy is even more marked in case the next iterate is found by the simple translation (8.22), according to Remark 36.

The effort of maintaining the model function $\phi_k(\boldsymbol{x})$ pays off when objective value and gradient estimation is taxing as compared to the re-solution of the model problem. If, moreover, the effort of gradient estimation is substantially larger than that of objective value estimation, then the line search (8.23) may prove more effective than the simple translation (8.22).

8.3 Summary

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The results recounted in Section 8.2 are my contribution.

I consider minimizing a convex objective function whose gradient computation is taxing, over a polyhedron. I propose a randomized version of the column generation scheme of Chapter 7, in an idealized setting, assuming that the objective function has bounded Hessians, and that unbiased gradient estimates of bounded variance can be constructed, with a reasonable effort.

I worked out a stochastic version of the unconstrained gradient descent method, and showed that it inherits the efficiency of the deterministic gradient descent, in case the objective function is uniformly well-conditioned throughout.

I developed a randomized column generation scheme, where new columns are found by the stochastic gradient descent method. I also include error analysis and reliability considerations.

The proposed method bears an analogy to stochastic gradient methods. The main difference is that the present method builds a model function. The effort of maintaining the model function pays off when objective value and gradient estimation is taxing as compared to the re-solution of the model problem.

Chapter 9

Handling a difficult constraint

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The new results presented in this chapter are my contribution.

I work out an approximation scheme for the solution of the convex constrained problem

$$\min \ \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to} \quad \breve{A}\boldsymbol{x} \leq \breve{\boldsymbol{b}}, \quad \phi(T\boldsymbol{x}) \leq \pi, \tag{9.1}$$

where the vectors $\boldsymbol{c}, \breve{\boldsymbol{b}}$ and the matrix $\breve{A}$ have compatible dimensions, and π is a given number.

The approximation scheme will apply the approach of Chapter 8. Like in that chapter, I work in an idealized setting, assuming that the function $\phi(\boldsymbol{z})$ has bounded Hessians. On the other hand, I do not exploit monotonicity of $\phi(\boldsymbol{z})$, hence the problems and models will be formulated according to (8.1 - 8.7).

The proposed scheme consists of the solution of a sequence of problems of the form (7.6), with an ever tightening stopping tolerance. We consider the linear constraint set $A\boldsymbol{x} \leq \boldsymbol{b}$ of problem (7.6). The last constraint of this set is $\boldsymbol{a}^r\boldsymbol{x} \leq b_r$, where $\boldsymbol{a}^r$ denotes the rth row of A, and b_r denotes the rth component of $\boldsymbol{b}$. Assume that this last constraint is a cost constraint, and let $\boldsymbol{c}^T = \boldsymbol{a}^r$ denote the cost vector. We consider a parametric form of the cost constraint, namely, $\boldsymbol{c}^T\boldsymbol{x} \leq d$, where $d \in \mathbb{R}$ is a parameter.

Let $\breve{A}$ denote the matrix obtained by omitting the rth row in A, and let $\breve{\boldsymbol{b}}$ denote the vector obtained by omitting the rth component in $\boldsymbol{b}$. Using these objects, we consider the problem

$$\min \ \phi(T\boldsymbol{x}) \quad \text{subject to} \quad \breve{A}\boldsymbol{x} \leq \breve{\boldsymbol{b}}, \quad \boldsymbol{c}^T\boldsymbol{x} \leq d, \tag{9.2}$$

with the parameter $d \in \mathbb{R}$. This parametric form of the unconstrained problem will be denoted by (7.6: $b_r = d$).

Let $\chi(d)$ denote the optimal objective value of problem (9.2), as a function of the parameter d. This is obviously a monotone decreasing convex function.

Let $\mathcal{I} \subset \mathbb{R}$ denote the domain over which the function is finite. We have either $\mathcal{I} = \mathbb{R}$ or $\mathcal{I} = [\underline{d}, +\infty)$ with some $\underline{d} \in \mathbb{R}$. Using the notation of the unconstrained problem, we say that $\chi(d)$ is the optimum of (7.6: $b_r = d$) for $d \in \mathcal{I}$.

Coming to the constrained problem (9.1), we may assume $\pi \in \chi(\mathcal{I})$. Let $d^\star \in \mathcal{I}$ be a solution of the equation $\chi(d) = \pi$, and let $l^\star(d)$ denote a linear support function to $\chi(d)$ at $d^\star$. In this section we work under

Assumption 40 *The support function $l^\star(d)$ has a significant negative slope, i.e., $l^{\star\prime} \ll 0$.*

It follows that the optimal objective value of (9.1) is $d^\star$.

Remark 41 *Assumption 40 is reasonable if the right-hand value π has been set by an expert, on the basis of preliminary experimental information. (A near-zero slope $l^{\star\prime}$ means that a slight relaxation of the probabilistic constraint allows a significant cost reduction.)*

We find a near-optimal $\widehat{d} \in \mathcal{I}$ using an approximate version of Newton's method. – The idea of regulating tolerances in such a procedure goes back to the Constrained Newton Method of Lemaréchal, Nemirovski and Nesterov [99]. Based on the convergence proof of the Constrained Newton Method, a simple convergence proof of Newton's method was reconstructed in [56]. I adapt the latter to the present case.

First, I describe a deterministic approximation scheme. Then, a randomized version is worked out.

9.1 A deterministic approximation scheme

Let the function $\phi(\boldsymbol{z})$ have bounded Hessians, as formulated in Assumption 30. In this section we work with exact function data, as formulated in

Assumption 42 *Given $\boldsymbol{z} \in \mathbb{R}^n$, the function value $\phi(\boldsymbol{z})$ and the gradient vector $\nabla\phi(\boldsymbol{z})$ can be computed exactly.*

A sequence of unconstrained problems (7.6: $b_r = d_\ell$) $(\ell = 1, 2, \ldots)$ is solved with increasing accuracy. In the course of this procedure, we build a single model $\phi_k(\boldsymbol{z})$ of the nonlinear objective $\phi(\boldsymbol{z})$, i.e., k is ever increasing. Columns added in the course of the solution of (7.6: $b_r = d_\ell$) are retained in the model and reused in the course of the solution of (7.6: $b_r = d_{\ell+1}$).

Given the ℓth iterate $d_\ell \in \mathcal{I}$, we need to estimate $\chi(d_\ell)$ with a prescribed accuracy. This is done by performing a column generation scheme with the master problem (8.6: $b_r = d_\ell$). Let $\overline{\mathcal{B}}_\ell$ denote an upper bound on the gap between the respective optima of the model problem (8.6: $b_r = d_\ell$) and the convex problem (7.6: $b_r = d_\ell$). Such a bound is constructed according to the expression (8.25). (In the present setup, it is a deterministic bound.)

Let moreover $\overline{\chi}_\ell$ denote the optimum of the model problem. With these objects we have

$$\overline{\chi}_\ell \;\geq\; \chi(d_\ell) \;\geq\; \overline{\chi}_\ell - \overline{\mathcal{B}}_\ell. \tag{9.3}$$

The column generation process with the master problem (8.6: $b_r = d_\ell$) is terminated if $\overline{\chi}_\ell$ and $\overline{\mathcal{B}}_\ell$ satisfy a stopping condition, to be discussed below.

Let $d_0, d_1 \in \mathcal{I},\ d_0 < d_1 < d^\star$ be the starting iterates. – The sequence of the iterates will be strictly monotone increasing, and converging to $d^\star$ from below.

Near-optimality condition for the constrained problem

Given a tolerance ϵ $(\pi \gg \epsilon > 0)$, let $\widehat{d} \in \mathcal{I}$ be such that

$$\widehat{d} \leq d^\star \quad \text{and} \quad \chi(\widehat{d}\,) \leq \pi + \epsilon. \tag{9.4}$$

Let $\widehat{\boldsymbol{x}}$ be an optimal solution of (9.2: $d = \widehat{d}\,$). Then $\widehat{\boldsymbol{x}}$ is an ϵ-feasible solution of (9.1) with objective value $\widehat{d}$. Exact feasible solutions of (9.1) have objective values not less than $d^\star \geq \widehat{d}$.

Stopping condition for the unconstrained subproblem

Let δ $(0 < \delta \ll \frac{1}{2})$ denote a fixed tolerance. (We can set e.g. $\delta = 0.25$ for the whole process.)

Given iterate $d_\ell \in \mathcal{I},\ d_\ell \leq d^\star$, we perform a column generation scheme with the master problem (8.6: $b_r = d_\ell$). The process is terminated if either

$$\begin{aligned} &(i) \qquad \overline{\chi}_\ell - \pi \;\leq \epsilon, \quad \text{or} \\ &(ii) \qquad \overline{\mathcal{B}}_\ell \;\leq\; \delta\,(\overline{\chi}_\ell - \pi) \end{aligned} \tag{9.5}$$

holds. Taking into account (9.3), we conclude:

If *(i)* occurs then $\widehat{d} := d_\ell$ satisfies the near-optimality condition (9.4), and the Newton-like procedure stops.

If *(ii)* occurs then $\overline{\chi}_\ell$ satisfies

$$\overline{\chi}_\ell \;\geq\; \chi(d_\ell) \;\geq\; \overline{\chi}_\ell - \delta\,(\overline{\chi}_\ell - \pi). \tag{9.6}$$

A new iterate will be constructed in the latter case.

Finding successive iterates

Given $\ell \geq 1$, assume that we have bounded $\chi(d_{\ell-1})$ and $\chi(d_\ell)$, as in (9.6). The graph of the function $\chi(d)$ is shown in Figure 9.1. Thick segments of the vertical lines $d = d_{\ell-1}$ and $d = d_\ell$ indicate confidence intervals – of the form (9.6) – for the function values $\chi(d_{\ell-1})$ and $\chi(d_\ell)$, respectively. Let $l_\ell : \mathbb{R} \to \mathbb{R}$ be the

linear function determined by the upper endpoint of the former interval, and the lower endpoint of the latter one. Formally,

$$l_\ell(d_{\ell-1}) := \overline{\chi}_{\ell-1} \ \geq \chi(d_{\ell-1}) \quad \text{and} \quad l_\ell(d_\ell) := \overline{\chi}_\ell - \delta\,(\overline{\chi}_\ell - \pi) \ \leq \chi(d_\ell), \quad (9.7)$$

where the inequalities follow from (9.6).

Due to the convexity of $\chi(d)$ and to Assumption 40, the linear function $l_\ell(d)$ obviously has a negative slope $l'_\ell \leq l^{\star\prime} \ll 0$. Moreover $l_\ell(d) \leq \chi(d)$ holds for $d_\ell \leq d$.

The next iterate $d_{\ell+1}$ will be the point satisfying $l_\ell(d_{\ell+1}) = \pi$. Of course $d_\ell < d_{\ell+1} \leq d^\star$ follows from the observations above.

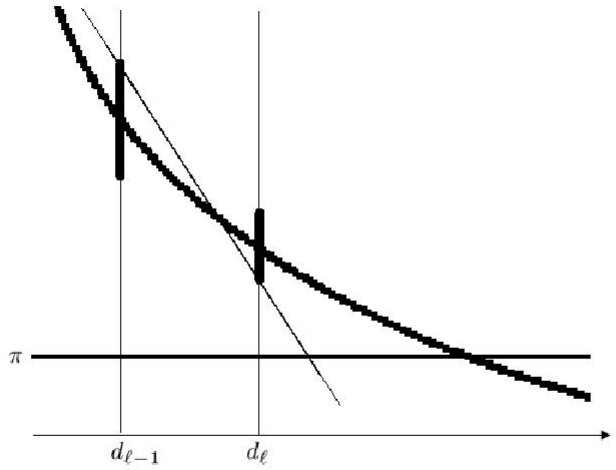

Figure 9.1: The graph of the function $\chi(d)$, and the construction of the next iterate.

Convergence

Let the iterates $d_0, d_1, \ldots, d_s$ and the linear functions $l_1(d), \ldots, l_s(d)$ be as defined above. We assume that $s > 1$, and the procedure did not stop before step $(s+1)$. Then we have

$$\overline{\chi}_\ell - \pi > \epsilon \quad (j = 0, 1, \ldots, s). \quad (9.8)$$

To simplify the notation, we introduce the linear functions $L_\ell(d) := l_\ell(d) - \pi \ \ (j = 1, \ldots, s)$. With these, (9.7) transforms into

$$L_\ell(d_{\ell-1}) = \overline{\chi}_{\ell-1} - \pi \quad \text{and} \quad L_\ell(d_\ell) \ = \ (1-\delta)(\overline{\chi}_\ell - \pi) \qquad (j = 1, \ldots, s). \quad (9.9)$$

Positivity of the above function values follows from (9.8). Moreover, the derivatives satisfy

$$L'_\ell \;=\; l'_\ell \;\le\; l^{\star\prime} \;\ll 0 \qquad (j = 1, \ldots, s) \tag{9.10}$$

due to the observations in the previous section.

Theorem 43 *We have*

$$\gamma^{s-1} \cdot \frac{|L'_1|}{|l^{\star\prime}|} \cdot L_1(d_1) \ge\; L_s(d_s) \qquad \textit{with} \quad \gamma := \left(\frac{1}{2(1-\delta)}\right)^2. \tag{9.11}$$

Proof. The following statements hold for $j = 1, \ldots, s-1$. From (9.9), we get

$$\frac{L_{\ell+1}(d_\ell)}{L_\ell(d_\ell)} \;=\; \frac{\overline{\chi}_\ell - \pi}{(1-\delta)(\overline{\chi}_\ell - \pi)} \;=\; \frac{1}{1-\delta}. \tag{9.12}$$

By definition, we have

$$L_\ell(d_\ell) + (d_{\ell+1} - d_\ell)\, L'_\ell \;=\; L_\ell(d_{\ell+1}) \;= 0.$$

It follows that $d_{\ell+1} - d_\ell = \frac{L_\ell(d_\ell)}{|L'_\ell|}$. Using this, we get

$$L_{\ell+1}(d_\ell) \;=\; L_{\ell+1}(d_{\ell+1}) + (d_\ell - d_{\ell+1})\, L'_{\ell+1} \;=\; L_{\ell+1}(d_{\ell+1}) + \frac{L_\ell(d_\ell)}{|L'_\ell|}\, |L'_{\ell+1}|.$$

Hence

$$\frac{L_{\ell+1}(d_\ell)}{L_\ell(d_\ell)} \;=\; \frac{L_{\ell+1}(d_{\ell+1})}{L_\ell(d_\ell)} + \frac{|L'_{\ell+1}|}{|L'_\ell|}. \tag{9.13}$$

From (9.12), we have

$$\frac{1}{1-\delta} \;=\; \frac{L_{\ell+1}(d_{\ell+1})}{L_\ell(d_\ell)} + \frac{|L'_{\ell+1}|}{|L'_\ell|} \;\ge\; 2\sqrt{\frac{L_{\ell+1}(d_{\ell+1})\,|L'_{\ell+1}|}{L_\ell(d_\ell)\,|L'_\ell|}}.$$

(This is the well-known inequality between means.) It follows that

$$\left(\frac{1}{2(1-\delta)}\right)^2 L_\ell(d_\ell)\,|L'_\ell| \;\ge\; L_{\ell+1}(d_{\ell+1})\,|L'_{\ell+1}|. \tag{9.14}$$

By induction, we get

$$\left(\frac{1}{2(1-\delta)}\right)^{2(s-1)} L_1(d_1)\,|L'_1| \;\ge\; L_s(d_s)\,|L'_s|. \tag{9.15}$$

Applying $|L'_s| \ge |l^{\star\prime}|$ we obtain (9.11). □

Example 44 *Let* $\delta = 0.25$, *then* $\gamma = \left(\frac{1}{2(1-\delta)}\right)^2 < 0.5$.

Corollary 45 *With the setting of Example 44, the number of Newton-like steps needed to reach the stopping tolerance ϵ does not exceed*

$$N(\epsilon) = \log\left(\frac{|L_1'|}{|l^{\star\prime}|} \cdot \frac{L_1(d_1)}{\epsilon}\right). \tag{9.16}$$

Note that $|l^{\star\prime}| \gg 0$ due to Assumption 40.

Given a problem, let us consider the efforts of its approximate solution as a function of the prescribed accuracy. That is on the order of $\log \frac{1}{\epsilon}$.

9.2 A randomized version of the approximation scheme

Let the function $\phi(\boldsymbol{z})$ have bounded Hessians, as formulated in Assumption 30. Let moreover Assumption 33 on the construction of unbiased gradient estimates having bounded variance hold.

Concerning the function $\chi(d)$, let Assumption 40 hold. Our aim, in principle, is the same as it has been in the deterministic case: find $\widehat{d} \in \mathcal{I}$ such that $\pi + \epsilon \geq \chi(\widehat{d}) \geq \pi$ holds with a pre-set tolerance ϵ. In the present uncertain environment, however, we may have to content ourselves with $\hat{d}$ such that $\pi + \epsilon \geq \chi(\hat{d}) > \pi - \epsilon$ holds. This problem statement is justifiable if the function $\chi(d)$ is not constant for $d > d^\star$. Let Assumption 46, below, hold.

Assumption 46 *There exists (an unknown) $d_\epsilon^\star \in \mathcal{I}$ such that $\chi\left(d_\epsilon^\star\right) = \pi - \epsilon$.*

Let q ($0.5 \ll q < 1$) denote a pre-set reliability. Using the randomized column generation scheme, a sequence of unconstrained problems (7.6: $b_r = d_\ell$) ($\ell = 1, 2, \ldots$) is solved, each with reliability q, and with an accuracy determined by the Newton-like approximation scheme. As in the deterministic case, we build a single model $\phi_k(\boldsymbol{z})$ of the nonlinear objective $\phi(\boldsymbol{z})$, i.e., k is ever increasing. Let $k_{\ell-1}$ denote the number of columns at the outset of the solution of problem (7.6: $b_r = d_\ell$).

Given the ℓth iterate $d_\ell \in \mathcal{I}$, we estimate $\chi(d_\ell)$ by performing a column generation scheme with the master problem (8.6: $b_r = d_\ell$). Applying the procedure of Chapter 8, we obtain an estimate $\overline{\mathcal{B}}_\ell$ for the gap between the respective optima of the model problem (8.6: $b_r = d_\ell$) and the convex problem (7.6: $b_r = d_\ell$). Keeping to the setting of Example 39, we set the reliability parameter to $q = 0.9$, obtaining $\mathrm{P}\left(\overline{\mathcal{B}}_\ell \geq \text{'gap'}\right) \geq 0.9$. (Note that the columns with indices up to $k_{\ell-1}$ belong to the initial model, hence in terms of Chapter 8, we have $\iota = k_{\ell-1}$.)

Let moreover $\overline{\chi}_\ell$ denote the optimum of the model problem. With these objects we have

$$\overline{\chi}_\ell \geq \chi(d_\ell) \quad \text{and} \quad \mathrm{P}\left(\chi(d_\ell) \geq \overline{\chi}_\ell - \overline{\mathcal{B}}_\ell\right) \geq 0.9. \tag{9.17}$$

We proceed in accordance with the deterministic scheme. The present stochastic scheme actually coincides with the deterministic one, provided the gap is estimated correctly in the unconstrained problem. In the stochastic scheme, however, we may underestimate the gap, meaning that $\overline{\mathcal{B}}_\ell$ is not an upper bound.

Consequently the inequality $\chi(d_\ell) \geq \overline{\chi}_\ell - \overline{\mathcal{B}}_\ell$ may not hold in (9.17). In such a case, $d_{\ell+1} > d^\star$ and hence $\overline{\chi}_{\ell+1} < \pi$ may occur. If the latter is observed, then we step back to the previous iterate, i.e., set $d_{\ell+2} = d_\ell$. We then carry on with the Newton-like procedure; first resolving the model problem (8.6: $b_r = d_{\ell+2}$) with reliability $q = 0.9$.

Stopping condition for the unconstrained subproblem

In accordance with the above discussion, we now formulate the stopping condition of the column generation process at the Newton-like step ℓ. Solution with the master problem (8.6: $b_r = d_\ell$) is terminated if $\overline{\chi}_\ell$ and $\overline{\mathcal{B}}_\ell$ satisfy one of the following conditions:

$$\begin{aligned} &(\alpha) \qquad \overline{\chi}_\ell < \pi, \\ &(\beta) \qquad \pi \leq \overline{\chi}_\ell < \pi + \epsilon \quad \text{and} \quad \overline{\mathcal{B}}_\ell \leq \epsilon, \\ &(\gamma) \qquad \pi + \epsilon \leq \overline{\chi}_\ell \quad \text{and} \quad \overline{\mathcal{B}}_\ell \ \leq \ \delta\,(\overline{\chi}_\ell - \pi). \end{aligned} \tag{9.18}$$

If condition (α) occurs, then we step back to the previous iterate $d_{\ell-1}$.

If condition (β) occurs, then we stop the Newton-like process.

If condition (γ) occurs, then we carry on to a new iterate $d_{\ell+1} > d_\ell$, like we did in the deterministic scheme.

Remark 47 *The stopping tolerance prescribed for the unconstrained subproblems is ever tightening in accordance with the progress of the Newton-like approximation scheme. However, the prescribed tolerance is never tighter than* $\delta \cdot \epsilon = 0.25\epsilon$.

Convergence and reliability

Let the unconstrained subproblems each be solved with a reliability of $q = 0.9$, and let δ, γ be set according to Example 44. Moreover, let us assume that the randomized Newton-like scheme did not stop in L steps. The aim of this section is to show that, provided L is large enough, an ϵ-optimal solution of the constrained problem has been reached with a high probability.

According to our assumption, case (β) did not occur in the stopping condition of the previous section. Let us define 'correct' and 'incorrect' steps, depending on the starting point d_ℓ:

– In case $d_\ell \leq d^\star$:
  We call step ℓ correct if $d_{\ell+1} \leq d^\star$ and $0.5 \cdot L_\ell(d_\ell)\,|L'_\ell| \geq L_{\ell+1}(d_{\ell+1})\,|L'_{\ell+1}|$ also holds, otherwise we call step ℓ incorrect.

– In case $d_\ell > d^\star$:
  We call step ℓ correct if a backstep occurs (i.e., if $d_{\ell+1} = d_{\ell-1}$), otherwise we call it incorrect.

A step is correct with a probability at least $q = 0.9$; this follows from the proof of Theorem 43, namely the expression (9.14).

If the difference between the number of the correct steps and the number of the incorrect steps exceeds $N(\epsilon)$, then an ϵ-optimal solution of the constrained problem has been reached, according to Corollary 45.

Let Z_ℓ be the random variable

$$Z_\ell = \begin{cases} 0 & \text{if step } \ell \text{ is correct,} \\ & \\ 1 & \text{if step } \ell \text{ is incorrect} \end{cases} \qquad (\ell = 1, \ldots, L).$$

As a step is correct with a probability at least $q = 0.9$, we have $\mathrm{E}(Z_\ell) \leq 0.1$, and hence $\mathrm{E}\left(\sum_{\ell=1}^{L} Z_\ell\right) \leq 0.1L$.

The difference between the number of the correct steps and the number of the incorrect steps is $L - 2\sum_{\ell=1}^{L} Z_\ell$. In order to show that the difference likely exceeds $N(\epsilon)$, we need an upper bound on the probability that $\sum_{\ell=1}^{L} Z_\ell$ is significantly larger than $\mathrm{E}\left(\sum_{\ell=1}^{L} Z_\ell\right)$.

Though all the gradient estimates were generated independently, there may be some interdependence among the random variables $Z_1, \ldots, Z_L$, because of the time structure of the process. But this interdependence is weak in the following sense. Suppose that we are at the beginning of the process. Given $0 < k \leq L$, we know that step k will be correct with a probability at least 0.9, no matter what happens in steps $1, \ldots, k-1$. In particular,

$$\mathrm{P}\left[\, Z_k = 1 \mid Z_\ell = 1 \, (\ell \in \mathcal{I}_k) \,\right] \leq 0.1 \quad \text{holds for all } \mathcal{I}_k \subseteq \{1, \ldots, k-1\}. \quad (9.19)$$

The condition in the above probability represents the event that $Z_\ell = 1$ occurs for every $\ell \in \mathcal{I}_k$. In case $k = 1$, the condition is empty, and (9.19) reduces to $\mathrm{P}(Z_1 = 1) \leq 0.1$.

Generalized Chernoff-Hoeffding bounds were proposed by Panconesi and Srinivasan in [121]. Intuitive proofs of such bounds, based on a simple combinatorial argument, were given by Impagliazzo and Kabanets in [84]. Recently, Pelekis and Ramon in [122] established a more general bound. I'm going to use a generalized Chernoff-type bound proposed by Panconesi and Srinivasan, in the form in which it was stated and proved by Impagliazzo and Kabanets:

Theorem 48 (Theorem 1.1 in [84]) *Let $Z_1, \ldots, Z_n$ be Boolean random variables such that, for some $p \in [0, 1]$,*

$$P\left[\, Z_\ell = 1 \, (\ell \in A) \,\right] \;\leq\; p^{|A|} \quad \textit{holds for all } \; A \subseteq \{1, \ldots, n\}, \qquad (9.20)$$

where $|A|$ denotes the cardinality of A.

Then, for any $\kappa \in [p, 1]$, we have

$$P\left[\sum_{\ell=1}^{n} Z_\ell \;\geq \kappa n\right] \;\leq\; e^{-nD(\kappa||p)}, \qquad (9.21)$$

where $D(.||.)$ is the relative entropy function, satisfying $D(\kappa||p) \geq 2(\kappa - p)^2$.

It is easy to see that our objects satisfy the precondition (9.20) with $p = 0.1$. Indeed, it follows from the repeated application of (9.19). – A formal proof may apply induction on n. For $n = 1$, we have $\mathrm{P}(Z_1 = 1) \leq 0.1$. Now let us assume that (9.20) holds for $1 \leq n < k$. The statement for $n = k$ follows from (9.19), by setting $\mathcal{I}_k = A \cap \{1, \ldots, k-1\}$.

As the precondition of Theorem 48 holds, we have (9.21) with $n = L$, $p = 0.1$ and $\kappa = 1/3$. Simple computation shows that, for $L \geq 22$,

$$\mathrm{P}\left[\sum_{\ell=1}^{L} Z_\ell \ < \frac{1}{3}\, L\right] \ \geq 0.9. \tag{9.22}$$

As we have seen, the difference between the number of the correct steps and the number of the incorrect steps is $L - 2\sum_{\ell=1}^{L} Z_\ell$ which exceeds $L/3$ if $\sum_{\ell=1}^{L} Z_\ell < L/3$ in (9.22) holds. I sum up the discussion in

Theorem 49 *Let the unconstrained problems each be solved with a reliability of $q = 0.9$; let δ, γ be set according to Example 44; and let*

$$L = \max\{\, 22,\, 3N(\epsilon)\,\}$$

with $N(\epsilon)$ defined in Corollary 45.

Assume that the randomized Newton-like scheme did not stop in L steps. Then an ϵ-optimal solution of the constrained problem has been reached with a probability at least 0.9.

Remark 50 *If case (β) occurred in the stopping condition of the previous section, then further checks are needed to ensure reliability.*

9.3 Summary

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The new results presented in this chapter are my contribution.

To handle a difficult constraint, I proposed a scheme that consists of the solution of a sequence of unconstrained problems with an ever tightening stopping tolerance. I adapt an approximate version of Newton's method to solving the problem sequence. – The idea of regulating tolerances in such a procedure goes back to the Constrained Newton Method of Lemaréchal, Nemirovski and Nesterov [99]. Based on the convergence proof of the Constrained Newton Method, a simple convergence proof of Newton's method was reconstructed in [56].

I worked out an approximation scheme that uses confidence intervals instead of function values. Based on this, I developed a randomized version. I include convergence analysis and reliability considerations.

Chapter 10

Adapting the randomized method to probability maximization

In this chapter I adapt the randomized approach of Chapter 8 to probability maximization. In this context, I look on Corollary 37 merely as a means of justification of the efficiency of the procedure. In order to measure the gap between the respective optima of the model problem and the original probabilistic problem, I'm going to propose a bounding approach.

The objective function will be $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ (or a regularized form), where $F(\boldsymbol{z})$ is an n-dimensional nondegenerate standard normal distribution function. Gradient estimates can be constructed with a reasonable effort, applying the simulation methods overviewed in Chapter 7. For the sake of simplicity, I assume that objective values are computed with a high precision. (In the present case of normally distributed random parameters, gradient computation is the bigger challenge. High-precision computation of a single non-zero component of the gradient requires an effort comparable to that of the objective value.)

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. Tamás Szántai is professor emeritus at the Budapest University of Technology and Economics. The rest of the coauthors are my colleagues at the John von Neumann University. The methodological results proved in Section 10.1 are my contribution. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues, developing and testing practical means of regulating accuracy and practical stopping conditions.

10.1 Reliability considerations

We solve the probability maximization problem (7.6). We work under Assumption 29: a feasible point $\check{\boldsymbol{z}}$ is known such that $F(\check{\boldsymbol{z}}) \geq 0.5$.

A bounded formulation

Exploiting monotonicity of the function $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$, the probability maximization problem with variable splitting can be formulated applying inequality between $\boldsymbol{z}$ and $T\boldsymbol{x}$, like it was done in Chapter 7. For convenience, I copy (7.7):

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \ \boldsymbol{z} - T\boldsymbol{x} \leq \boldsymbol{0}. \tag{10.1}$$

A further speciality of the normal distribution function is the existence of a bounded box $\mathcal{Z}$ outside which the probability weight can be ignored. Including the constraint $\boldsymbol{z} \in \mathcal{Z}$ in (10.1) results in a closely approximating problem:

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \ \boldsymbol{z} - T\boldsymbol{x} \leq \boldsymbol{0},\ \ \boldsymbol{z} \in \mathcal{Z}. \tag{10.2}$$

Observation 51 *The difference between the respective optima of problems (10.1) and (10.2) is insignificant.*

Proof. Let $\boldsymbol{z}$ be a part of a feasible solution of (10.1), and let us consider the box $(\boldsymbol{z} + \mathcal{N}) \cap \mathcal{Z}$, where $\mathcal{N}$ denotes the negative orthant.

In case this box is empty, we have $F(\boldsymbol{z}) \approx 0$ due to the specification of $\mathcal{Z}$. Taking into account Assumption 29, such $\boldsymbol{z}$ cannot be a part of an optimal solution of (10.1).

In case the box $(\boldsymbol{z} + \mathcal{N}) \cap \mathcal{Z}$ is not empty, let $\Pi_{\boldsymbol{z}}$ denote its 'most positive' vertex. We have $\Pi_{\boldsymbol{z}} \in \mathcal{Z}$, $\Pi_{\boldsymbol{z}} \leq \boldsymbol{z}$, and $F(\Pi_{\boldsymbol{z}}) \approx F(\boldsymbol{z})$. If $F(\boldsymbol{z}) < 0.5$, then, due to Assumption 29 again, $\boldsymbol{z}$ cannot be a partial optimal solution of (10.1).

In the remaining case of $F(\Pi_{\boldsymbol{z}}) \approx F(\boldsymbol{z}) \geq 0.5$, we have $\phi(\Pi_{\boldsymbol{z}}) \approx \phi(\boldsymbol{z})$. Moreover $\Pi_{\boldsymbol{z}}$ is a partial feasible solution of (10.2), due to $\Pi_{\boldsymbol{z}} \in \mathcal{Z}$, $\Pi_{\boldsymbol{z}} \leq \boldsymbol{z}$. □

I assume that the known feasible point $\check{\boldsymbol{z}}$ with $F(\check{\boldsymbol{z}}) \geq 0.5$ falls into $\mathcal{Z}$. (Otherwise we can consider its projection to $\mathcal{Z}$.) Hence $\check{\boldsymbol{z}}$ is a feasible solution of (10.2).

Remark 52 *We could base the construction of Chapter 7 on (10.2), instead of (7.7). Formally, this would mean working with the restricted functions*

$$\phi_{\mathcal{Z}}(\boldsymbol{z}) = \begin{cases} \phi(\boldsymbol{z}) & \text{if } \boldsymbol{z} \in \mathcal{Z}, \\ +\infty & \text{otherwise} \end{cases} \qquad \text{and} \qquad \phi^{\star}_{\mathcal{Z}}(\boldsymbol{u}) = \max_{\boldsymbol{z} \in \mathcal{Z}} \{\boldsymbol{u}^T \boldsymbol{z} - \phi(\boldsymbol{z})\} \tag{10.3}$$

instead of $\phi(\boldsymbol{z})$ and $\phi^{\star}(\boldsymbol{u})$, respectively.

In a pure form of this bounded scheme, new columns are always selected from $\mathcal{Z}$. An obvious drawback is that Theorem 31 does not apply to the resulting bounded optimization problem.

I presently develop a hybrid scheme, including a restriction to $\mathcal{Z}$ in the master problem, but selecting new columns by unconstrained maximization.

A hybrid form of the column generation scheme

Introducing new variables $\boldsymbol{z}' \in \mathbb{R}^n$, we transform (10.2) to

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \ \boldsymbol{z}' - T\boldsymbol{x} \leq \boldsymbol{0},\ \ \boldsymbol{z}' \in \mathcal{Z},\ \ \boldsymbol{z} \leq \boldsymbol{z}'. \quad (10.4)$$

The above problem has the general pattern of (7.7), hence the dual problem can be formulated in the manner of Chapter 7. Model problems are then formulated accordingly.

Let the feasible point $\check{\boldsymbol{z}}$ with $F(\check{\boldsymbol{z}}) \geq 0.5$ be included among the initial $\boldsymbol{z}_i$ $(i = 0, \ldots, k)$ testpoints. Then the common optimum of the model problems will never exceed $-\log 0.5$. It follows that the current $\overline{\boldsymbol{z}}$, obtained in the form (7.16) with an optimal solution of the model problem, will always satisfy $F(\overline{\boldsymbol{z}}) \geq 0.5$ and $\overline{\boldsymbol{z}} \in \mathcal{Z}$.

Let $\overline{\boldsymbol{g}} = \nabla\phi(\overline{\boldsymbol{z}})$ be the corresponding gradient. Let moreover $(\,\overline{\vartheta},\, \overline{\boldsymbol{u}}\,)$ be part of an optimal dual solution of the current model problem. Finally, let $\overline{\mathcal{R}}$ denote the gap between the respective optima of the model problem and the original probabilistic problem.

Observation 53 *With the above objects, we have:*

$$\overline{\mathcal{R}} \leq \ \Big(\phi_k(\overline{\boldsymbol{z}}) - \phi(\overline{\boldsymbol{z}})\Big) + \max_{\boldsymbol{z}\in\mathcal{Z}} (\overline{\boldsymbol{u}} - \overline{\boldsymbol{g}})^T(\boldsymbol{z} - \overline{\boldsymbol{z}}). \quad (10.5)$$

Proof. An adaptation of Observation 38 to the present bounded setting is

$$\overline{\mathcal{R}} = \ \max_{\boldsymbol{z}\in\mathcal{Z}} \left\{\, \overline{\vartheta} + \overline{\boldsymbol{u}}^T\boldsymbol{z} - \phi(\boldsymbol{z})\right\}.$$

Taking into account that $\overline{\vartheta} = \phi_k(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}^T\overline{\boldsymbol{z}}$ according to Observation 27 *(a)*, and that $\phi(\boldsymbol{z}) \geq \phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{g}}^T(\boldsymbol{z} - \overline{\boldsymbol{z}})$ holds due to the convexity of $\phi(\boldsymbol{z})$, we obtain (10.5). □

The exact gradient $\overline{\boldsymbol{g}}$ is of course not known, but we can construct a gradient estimate together with a confidence interval. Given an error tolerance $\Delta > 0$ and a probability p $(0 < p \ll 1)$, let $\overline{\boldsymbol{G}}$ and $\overline{\mathcal{I}}$ denote our gradient estimate and confidence interval, respectively. The interval has the vector as a center, and they satisfy the following rules:

$$\mathrm{E}(\overline{\boldsymbol{G}}) = \overline{\boldsymbol{g}}, \quad \mathrm{P}\left(\,\overline{\boldsymbol{g}} \in \overline{\mathcal{I}}\,\right) \geq 1 - p \quad \text{and} \quad \mathrm{diag}\left(\overline{\mathcal{I}}\right) \leq \Delta, \quad (10.6)$$

where diag denotes the largest distance in the interval.

The following observation shows that we can use $\overline{\boldsymbol{G}}$ to estimate the maximum on the right-hand side of (10.5).

Observation 54 *The objects of (10.6) admit the following estimate:*

$$\max_{\boldsymbol{z}\in\mathcal{Z}} (\overline{\boldsymbol{u}} - \overline{\boldsymbol{g}})^T(\boldsymbol{z} - \overline{\boldsymbol{z}}) \ \ \leq \ \ \max_{\boldsymbol{z}\in\mathcal{Z}} (\overline{\boldsymbol{u}} - \overline{\boldsymbol{G}})^T(\boldsymbol{z} - \overline{\boldsymbol{z}}) \ + \ \Delta \cdot \mathrm{diag}(\mathcal{Z}) \quad (10.7)$$

holds with a probability at least $1 - p$.

Proof. Based on the confidence interval, a pessimist estimate of the left-hand side of (10.7) could be obtained by solving the (nonconvex) quadratic programming problem

$$\max\, (\overline{\boldsymbol{u}} - \boldsymbol{g})^T (\boldsymbol{z} - \overline{\boldsymbol{z}}) \quad \text{such that} \quad \boldsymbol{z} \in \mathcal{Z},\ \ \boldsymbol{g} \in \overline{\mathcal{I}}. \tag{10.8}$$

Instead of the quadratic programming problem, we just solve the linear programming problem

$$\max\, (\overline{\boldsymbol{u}} - \overline{\boldsymbol{G}})^T (\boldsymbol{z} - \overline{\boldsymbol{z}}) \quad \text{such that} \quad \boldsymbol{z} \in \mathcal{Z}. \tag{10.9}$$

Denoting an optimal solution of (10.8) by $(\grave{\boldsymbol{z}}, \grave{\boldsymbol{g}})$, and an optimal solution of (10.9) by $\widehat{\boldsymbol{z}}$, the difference between the respective optima is

$$\begin{aligned} (\overline{\boldsymbol{u}} - \grave{\boldsymbol{g}})^T (\grave{\boldsymbol{z}} - \overline{\boldsymbol{z}}) - (\overline{\boldsymbol{u}} - \overline{\boldsymbol{G}})^T (\widehat{\boldsymbol{z}} - \overline{\boldsymbol{z}}) \ &\leq\ (\overline{\boldsymbol{u}} - \grave{\boldsymbol{g}})^T (\grave{\boldsymbol{z}} - \overline{\boldsymbol{z}}) - (\overline{\boldsymbol{u}} - \overline{\boldsymbol{G}})^T (\grave{\boldsymbol{z}} - \overline{\boldsymbol{z}}) \\ &=\ (\overline{\boldsymbol{G}} - \grave{\boldsymbol{g}})^T (\grave{\boldsymbol{z}} - \overline{\boldsymbol{z}}), \end{aligned}$$

where the inequality is a consequence of the selection of $\widehat{\boldsymbol{z}}$.

The Cauchy–Bunyakovsky–Schwarz inequality yields (10.7). □

I sum up the above discussion in

Corollary 55 *With the above-defined objects, the random quantity*

$$\overline{\mathcal{B}} := \Big(\phi_k(\overline{\boldsymbol{z}}) - \phi(\overline{\boldsymbol{z}})\Big) + \max_{\boldsymbol{z} \in \mathcal{Z}} (\overline{\boldsymbol{u}} - \overline{\boldsymbol{G}})^T (\boldsymbol{z} - \overline{\boldsymbol{z}}) \ +\ \Delta \cdot \mathrm{diag}(\mathcal{Z}) \tag{10.10}$$

is a probabilistic bound on the gap between the respective optima of the model problem and the original probabilistic problem, i.e., $P\big(\overline{\mathcal{B}} \geq \text{'gap'}\big) \geq 1 - p$ *holds.*

Regulating accuracy and reliability

From the efficiency point of view, Assumption 33 on the limited variance of the gradient estimates must be satisfied. From the reliability point of view, we need confidence intervals for the gradient vectors in the bounding approach.

Given iterate $\overline{\boldsymbol{z}}$, we wish to construct an estimate $\overline{\boldsymbol{G}}$ for the corresponding gradient. We have two objectives. On the one hand, we need Corollary 37 to ensure efficiency of a descent step in the course of the column selection. Hence (8.13) should hold with an appropriate σ between the vectors $\boldsymbol{g}^\circ = \overline{\boldsymbol{g}} - \overline{\boldsymbol{u}}$ and $\boldsymbol{G}^\circ = \overline{\boldsymbol{G}} - \overline{\boldsymbol{u}}$. Specifically,

$$\mathrm{E}\left(\left\|\overline{\boldsymbol{G}} - \overline{\boldsymbol{g}}\right\|^2\right) \ \leq\ \sigma^2 \left\|\overline{\boldsymbol{g}} - \overline{\boldsymbol{u}}\right\|^2 \quad \text{should hold.} \tag{10.11}$$

On the other hand, we need (10.6) to hold with appropriate parameters Δ and p to ensure that the bound $\overline{\mathcal{B}}$ is tight and reliable. I slightly re-formulate the definition of $\overline{\mathcal{B}}$ in (10.10) as follows:

$$\Big(\phi_k(\overline{\boldsymbol{z}}) - \phi(\overline{\boldsymbol{z}})\Big) + \max_{\boldsymbol{z} \in \overline{\mathcal{Z}}} \Big((\overline{\boldsymbol{u}} - \overline{\boldsymbol{g}}) - (\overline{\boldsymbol{G}} - \overline{\boldsymbol{g}})\Big)^T (\boldsymbol{z} - \overline{\boldsymbol{z}}) \ +\ \Delta \cdot \mathrm{diag}(\mathcal{Z}). \tag{10.12}$$

Concerning p, we increase reliability with each master iteration, as we did in the general case of Chapter 8. Having added κ columns, we prescribe the reliability $1 - p_\kappa$, with p_κ set according to Example 39.

In setting the parameters σ and Δ, we aim to find a balance between the error of the polyhedral model function on the one hand, and the error of the gradient estimation on the other hand. According to Observation 27 *(c)*, $\overline{\boldsymbol{u}} \in \partial\phi_k(\overline{\boldsymbol{z}})$ holds. Taking into account $\overline{\boldsymbol{g}} = \nabla\phi(\overline{\boldsymbol{z}})$, the vector $\overline{\boldsymbol{u}} - \overline{\boldsymbol{g}}$ in (10.11) and (10.12) represents the gradient error of the polyhedral model function $\phi_k(\boldsymbol{z})$. Similarly, $\phi_k(\overline{\boldsymbol{z}}) - \phi(\overline{\boldsymbol{z}})$ in (10.12) represents the error in function value. On the other hand, the vector $\overline{\boldsymbol{G}} - \overline{\boldsymbol{g}}$ in (10.11) and (10.12) represents the error of the gradient estimate $\overline{\boldsymbol{G}}$.

A balance between those two types of error is found by a two-stage procedure. We begin with estimating the order of the magnitude of $\|\overline{\boldsymbol{u}} - \overline{\boldsymbol{g}}\|$, and based on this, we decide the size of the sample to be used in gradient estimation. The simulation methods of Section 7.3 can be applied.

Remark 56 *The bound $\boldsymbol{z} \in \mathcal{Z}$ in (10.2) allows regularization of the objective function, in the form of $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{z}\|^2$ with $\rho > 0$. Substituting this regularized objective in (10.2) makes no significant variation in the objective value of $\boldsymbol{z} \in \mathcal{Z}$, provided ρ is small enough. The regularizing term improves the condition of the objective: eigenvalues of the Hessians are increased by ρ, hence $\nabla^2\phi(\boldsymbol{z}) \succeq \rho I$ $(\boldsymbol{z} \in \mathbb{R}^n)$ holds.*

On the other hand, the regularized objective will no longer be monotone, hence in (10.4), the constraint $\boldsymbol{z} \leq \boldsymbol{z}'$ must be changed to $\boldsymbol{z} = \boldsymbol{z}'$. The resulting problem has the general pattern of (8.1), hence the dual problem and the model problems can be formulated in the manner of Chapter 8.

10.2 A computational experiment

We implemented the procedure with the coauthors of [54]. The aim of this experiment is to demonstrate the workability of the randomized column generation scheme, in case of probabilistic problems. Namely, we have $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ with an n-dimensional nondegenerate normal distribution function $F(\boldsymbol{z})$.

Setup

Like in Chapter 7, we tested our implementation on a cash matching problem, with a fifteen dimensional normal distribution.

Our solver is based on the implementation described in Chapter 7. In this version we used the randomized procedure of Chapter 8 and implemented the hybrid form of the column generation scheme, as described in the present chapter. (The bounded box $\mathcal{Z}$ was set so that $\mathrm{P}(\mathcal{Z}) = 0.99$ holds.)

In the course of the randomized column generation scheme, we perform just a single line search in each column generation subproblem. This line search starts from the current $\overline{\boldsymbol{z}}$ vector.

Probabilistic function values are computed and gradients are estimated by the Genz subroutines, controlling accuracy through the sample size. In the present simple implementation of the iterative scheme, the distribution function values $F(\boldsymbol{z}_i)$ are always computed with a high accuracy, setting the sample size to $10,000$. On the other hand, gradients are estimated in such a way that the norm of the error of the current gradient $\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ be less than one tenth of the norm of the previous gradient $\|\nabla\phi(\overline{\boldsymbol{z}}_-) - \overline{\boldsymbol{u}}_-\|$.

Results and observations

We performed 10 runs of the randomized procedure, each with 50 iterations. The sequences of the probability levels obtained, i.e., of the values $F(\overline{\boldsymbol{z}})$, are shown in Figure 10.1. At each iteration, the gradient $\nabla\phi(\overline{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ is estimated by $\overline{\boldsymbol{G}} - \overline{\boldsymbol{u}}$. The norm of this estimate decreases as the procedure progresses. For a single typical run, this decrease is shown in Figure 10.2.

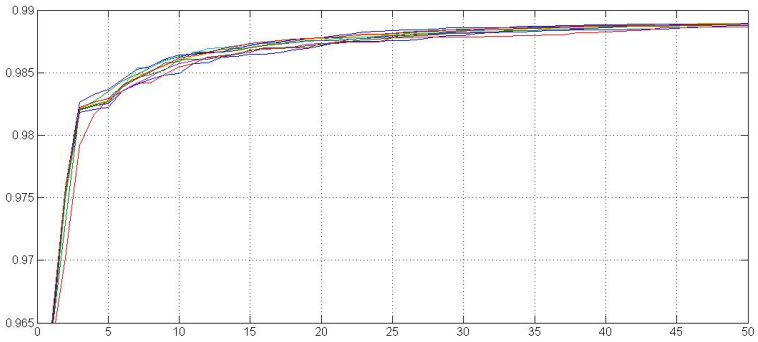

Figure 10.1: Probability levels obtained, as a function of iteration counts. Different runs are represented by different threads.

We applied no stopping condition besides iteration count. After 50 iterations, optimal probability levels obtained in the different runs were already very near to each other (the difference between highest and lowest being less than 0.0003.) On the other hand, the value of the bound $\overline{\mathcal{B}}$ of (10.10) was between 0.025 and 0.03 at the end of our runs. We conclude that, though the bounding procedure is workable, it needs further technical improvements to keep pace with the stochastic approximation scheme.

In accordance with the hybrid bounding form of Section 10.1, we did not restrict new columns $\boldsymbol{z}_i$ to the box $\mathcal{Z}$. Still, the probability level was high in all iterates, $F(\boldsymbol{z}_i) \geq 0.9$ holding with the columns added in the course of the column generation process. This allowed high-accuracy computation of all probabilistic function values. The restriction $\boldsymbol{z}' \in \mathcal{Z}$ of (10.4) was never active in any optimal solution of the master problem.

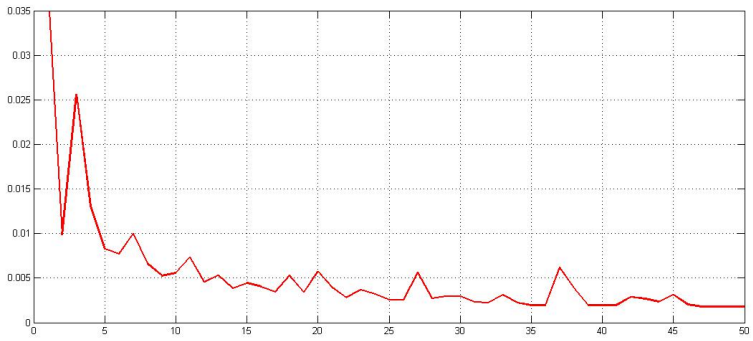

Figure 10.2: Decrease of the gradient norm as a function of iteration counts, in a single run.

10.3 Summary

This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. Tamás Szántai is professor emeritus at the Budapest University of Technology and Economics. The rest of the coauthors are my colleagues at the John von Neumann University. The methodological results proved in Section 10.1 are my contribution. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues, developing and testing practical means of regulating accuracy and practical stopping conditions.

The objective function is $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$, where $F(\boldsymbol{z})$ is an n-dimensional nondegenerate standard normal distribution function. Gradient estimates can be constructed with a reasonable effort, applying the simulation methods overviewed in Chapter 7. For the sake of simplicity, I assume that objective values are computed with a high precision. – In the present case of normally distributed random parameters, gradient computation is the bigger challenge. High-precision computation of a single non-zero component of the gradient requires an effort comparable to that of the objective value. (A means of alleviating the difficulty of gradient computation in case of multivariate normal distribution has recently been proposed in [75].)

The proposed method is an adaptation of the randomized approach of Chapter 8. In the probabilistic context, reliability considerations are based on a special bounding approach. We demonstrate the workability of the approach with a computational experiment.

In comparison with the outer approximation approach widely used in probabilistic programming, I mention that the latter is difficult to implement due

to noise in gradient computation. The outer approximation approach applies a direct cutting-plane method. Even a fairly accurate gradient may result in a cut cutting into the epigraph of the probabilistic function (especially in regions farther away from the current iterate). One either needs sophisticated tolerance handling to avoid cutting into the epigraph — e.g., Szántai [163], Mayer [108], Arnold et al. [5], — or else one needs a sophisticated convex optimization method that can handle cuts cutting into the epigraph — e.g., Oliveira et al. [28], Van Ackooij and Sagastizábal [175]. — Yet another alternative is perpetual adjustment of existing cuts to information revealed in the course of the process; e.g., Higle and Sen [81].

Inner approximation of the level set $\mathcal{L}_p$, the approach initiated by Prékopa [130], results in a model that is easy to validate. The level set is approximated by means of p-efficient points. In the cone generation approach initiated by Dentcheva, Prékopa and Ruszczyński [40], new approximation points are found by minimization over $\mathcal{L}_p$. As this entails a substantial computational effort, the master part of the decomposition framework should succeed with as few p-efficient points as possible. This calls for specialized solution methods like those of [37], [39], [173]. An increasing level of complexity is noticeable.

I proposed inner approximation of the epigraph of the probabilistic function. This approach admits easier generation of new test points, and endures noise in gradient computation without any special effort. Noisy gradient estimates may yield iterates that do not improve much on our current model. But we retain a reliable inner approximation of the function. This inherent stability of the model enables the application of randomized methods of simple structure.

The proposed stochastic approximation procedure can be implemented using standard components. The master problem is conveniently solved by an off-the-shelf solver. New approximation points are found through simple line search whose direction can be determined by standard implementations of classic Monte Carlo simulation procedures.

Chapter 11

A summary of the summaries

I deal with computational aspects of stochastic programming. I have collaborated with leading teams in computational stochastic programming and industrial optimization, helping them develop specialized solvers for different real-life problems. I have also coordinated research and development projects at the John von Neumann University, former Kecskemét College.

In the course of these projects I developed new versions of classic algorithms and combined classic algorithmic components in new ways. I examined theoretical efficiency of these algorithms. Most of these methods have been implemented and thoroughly tested. Several of them have been intensively applied.

Large amounts of data to be organized, and inaccuracy in function evaluations are characteristic features of stochastic programming problems. Similar solution approaches proved effective for very diverse problems; enhanced cutting-plane methods in primal and dual forms. Cutting-plane methods and enhancements have been discussed in Chapter 2. Chapters 3 - 6 deal with the adaptation of such methods to risk-averse and two-stage stochastic programming problems. Dual-form cutting plane methods, in the shape of column generation schemes, have been discussed in Chapters 7 - 10, with application to probabilistic programming problems.

Cutting-plane methods and enhancements (Chapter 2).
In [51], I developed approximate versions of the level method and the constrained level method of Lemaréchal, Nemirovskii and Nesterov [99]. I extended the original convergence proofs to the approximate methods. My approximate level method was one of the precursors of the 'on-demand accuracy' approach of the Charles Broyden Prize-winning paper of Oliveira and Sagastizábal [27].

In [64] and [184], I worked out a special version of the on-demand accuracy approach. In this dissertation, I call the resulting methods 'partially inexact'. I extended the on-demand accuracy approach to constrained problems.

My approximate and partially inexact methods have been successfully applied in the solution of diverse stochastic programming problems. The computational studies [62] and [184] indicate that my unconstrained versions inherit the superior experimental efficiency of the level method.

Van Ackooij and de Oliveira in [174] extended my partially inexact version of the constrained level method to handle upper oracles.

Cutting-plane methods for risk-averse problems (Chapter 3). The convex conjugacy relationship known to exist between expected shortfall and tail expectation, reduces to linear programming duality in case of discrete finite distributions, as I worked out in [52]. This approach yields a conditional value-at-risk formulation that proved effective for handling CVaR constraints in two-stage problems.

I worked out cutting-plane approaches for the handling of second-order stochastic dominance in stochastic programming problems. These were implemented and investigated in collaboration with Gautam Mitra and Diana Roman. Algorithmic descriptions and test results were presented in [57]. The cutting-plane approach resulted in dramatic improvement in efficiency.

I proposed a scaled version of the uniform-dominance model of Roman, Darby-Dowman, and Mitra [147]. We compared modeling aspects of the scaled and the unscaled dominance measures in collaboration with Gautam Mitra, Diana Roman and Victor Zverovich. Algorithmic descriptions and test results were presented in [58]. This study confirmed a shape-preserving quality of the scaled dominance measure. In a subsequent computational study [148], my former co-workers Roman, Mitra and Zverovich observe that the scaled model has a very good backtesting performance.

The models and solvers developed in the course of the projects [57] and [58] have been included in the optimization and risk analytics tools developed at OptiRisk Systems.

Enhanced versions of the cutting-plane method described in [57] were developed by Sun et al. [160] and Khemchandani et al. [88].

Decomposition methods for two-stage SP problems (Chapter 4). I adapted the approximate level method [51] (recounted in Chapter 2) to solve the aggregate master problem in a decomposition scheme. The inexact oracle was based on a successive approximation of the distribution. The novelty of the approach is that the accuracy of the distribution approximation is regulated by the optimization method applied to the master problem. This setup helps in finding a balance between different efforts. We implemented the method with Zoltán Szőke. The method was described and a computational study was presented in [62]. Our numerical results show that this approximation framework is workable. Our work influenced the projects of Oliveira, Sagastizábal and Scheimberg [119] and Song and Luedtke [157].

In the computational study [190] we re-assessed different solution approaches to two-stage problems, in view of recent advances in computer architecture and

LP solver algorithms. My co-workers were Gautam Mitra, Francis Ellison and Victor Zverovich. We demonstrated that decomposition methods generally scale better than direct solution of the equivalent LP problem. Moreover, we found that decomposition based on the aggregate model scales better than that based on the disaggregate model.

The decomposition-based solvers developed in the course of the project [190] have been included in the FortSP stochastic programming solver system of Opti-Risk. FortSP has been applied in diverse real-life application projects.

One of my co-workers was Victor Zverovich whose PhD dissertation [189] is based to a large extent on our joint project.

Our computational study influenced the team of A. Koberstein to develop decomposition methods to solve their application problems, as reported in [185]. Our findings served as guidelines to Gondzio et al. in their project [73], and were considered as reference by Takano et al. [167] and Sen and Liu [154].

I adapted my partially inexact version of the level method (recounted in Chapter 2), to two-stage stochastic programming problems. This method admits an intuitive oracle rule: no recourse problem is solved if the disaggregate model function value is significantly higher than the aggregate one, as evaluated at the new iterate. With this rule, the method combines the advantages of the traditional aggregate and disaggregate models. We implemented the method and performed an extensive computational study in collaboration with Leena Suhl, Achim Koberstein and Christian Wolf. Our test results demonstrate the efficiency of the approach. On average, it solved the test problems five times faster than traditional single-cut Benders that we considered benchmark.

The methods and solvers developed in the course of the projects [64] and [184] have been applied in the solution of a real-life problem of a gas utility company. One of my co-workers was Christian Wolf who wrote his PhD thesis [183] partly with me as advisor.

Feasibility issues in two-stage SP problems (Chapter 5).
In [62], I proposed handling second-stage infeasibility through a constraint function in the master problem, and adapted the approximate constrained level method [51] (recounted in Chapter 2) to solve the resulting special master problem. This approach is more effective then the application of feasibility cuts. The method is of the primal-dual type, and the rule of tuning the dual iterate keeps a fine balance between feasibility and optimality issues. Moreover, the regularization extends to feasibility issues.

But this approach requires extending the recourse function to the whole space. I worked with infeasibility-penalized formulations of the recourse problem. The necessary penalty is easily computable in the case of network recourse problems. (For general recourse problems, I worked out a means of adjusting the penalty parameter in the course of the solution process.)

We implemented the methods with Zoltán Szőke. Computational experiments confirmed the workability of the approach.

Risk constraints in two-stage SP problems (Chapter 6).
In [64], I generalized the on-demand accuracy approach to risk-averse two-stage problems. I considered two problem types, applying a CVaR constraint or a stochastic ordering constraint, respectively, on the recourse. I reformulated the latter problem using the dominance measure recounted in Chapter 3. I adapted the partially inexact version of the constrained level method (recounted in Chapter 2) to the resulting risk-averse problems.

In collaboration with Leena Suhl, Achim Koberstein and Christian Wolf, we implemented and compared different methods for the solution of the CVaR-constrained problem, and performed an extensive computational study. Each of the decomposition methods outperformed the direct solution approach in our experiments. The effect of regularization proved remarkable. For large problems the regularized on-demand accuracy approach proved most effective.

We formulated and solved large instances of the CVaR-constrained version of the real-life strategic gas purchase planning problem of my co-workers. The aim was to hedge against the risk caused by potential low demand. The gas utility company decided not to implement the optimal solution obtained from the risk-averse problems; they preferred an insurance cover. The experiments were still useful, because decision makers could compare the cost of an insurance cover to the decrease in average profit due to a risk constraint.

Probabilistic problems (Chapter 7).
In [55], I proposed a polyhedral approximation of the epigraph of the probabilistic function. I worked out a successive approximation scheme for probability maximization; new approximation points are added as the process progresses. This results in a column generation scheme with a linear programming master problem. (From a dual point of view, the column generation approach is a cutting-plane method applied to a conjugate function.) This project was motivated by the concept of p-efficient points proposed by Prékopa [130], and the approximation scheme is analogous to the cone generation scheme of Dentcheva, Prékopa and Ruszczyński [40]. But in my scheme, the subproblem of finding a new approximation point is easier; it is an unconstrained convex optimization problem, solvable by a simple gradient descent method. The approach is easy to implement and endures noise in gradient computation. Hence the classic probability estimation methods are applicable.

My coauthors were Edit Csizmás, Rajmund Drenyovszki, Wim van Ackooij, Tibor Vajnai, Lóránt Kovács and Tamás Szántai. The model problems and optimization methods were developed by me. Wim van Ackooij collaborated in compiling a historical overview, and in test problem selection. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues concerning the oracle and finding a starting solution.

Most of the computational effort in our tests was spent in gradient computation. In order to balance different efforts, we decided to apply rough approxi-

mate solutions for the column generation subproblems, performing just a single line search in each gradient descent method. This heuristic procedure never resulted in any substantial increase in the number of new columns needed to solve a test problem. I found this interesting and investigated possible causes.

The gradient descent method is remarkably effective in case the objective function is well-conditioned in a certain neighbourhood of the optimal solution. I make a case for conjecturing that the probabilistic function is well-conditioned in an area where potential optimal solutions typically belong. (The bounded formulation of Chapter 10 allows the regularization of a poorly conditioned objective.) The column generation procedure gains traction as an optimal solution is gradually approached.

A randomized method for a difficult objective (Chapter 8).
This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The new results presented in this chapter are my contribution.

I consider minimizing a convex objective function whose gradient computation is taxing, over a polyhedron. I propose a randomized version of the column generation scheme of Chapter 7, in an idealized setting, assuming that the objective function has bounded Hessians, and that unbiased gradient estimates of bounded variance can be constructed. I worked out a stochastic version of the unconstrained gradient descent method, and showed that it inherits the efficiency of the deterministic gradient descent, in case the objective function is uniformly well-conditioned throughout. I developed a randomized column generation scheme, where new columns are found by the stochastic gradient descent method. I also include error analysis and reliability considerations.

The proposed method bears an analogy to stochastic gradient methods. The main difference is that the present method builds a model function. The effort of maintaining the model function pays off when objective value and gradient estimation is taxing as compared to the re-solution of the model problem.

Handling a difficult constraint (Chapter 9).
This chapter is based on Fábián, Csizmás, Drenyovszki, Vajnai, Kovács and Szántai [54]. The new results presented in this chapter are my contribution.

To handle a difficult constraint, I proposed a scheme that consists of the solution of a sequence of unconstrained problems with an ever tightening stopping tolerance. I adapt an approximate version of Newton's method to solving the problem sequence. – The idea of regulating tolerances in such a procedure goes back to the Constrained Newton Method of Lemaréchal, Nemirovski and Nesterov [99]. – I worked out an approximation scheme that uses confidence intervals instead of function values. Based on this, I developed a randomized version. I include convergence analysis and reliability considerations.

Randomized maximization of probability (Chapter 10).
In this chapter, I proposed a randomized inner approximation scheme for probability maximization. The chapter is based on Fábián, Csizmás, Drenyovszki,

Vajnai, Kovács and Szántai [54]. The methodological results proved in Section 10.1 are my contribution. Tamás Szántai contributed with his expertise in estimating distribution function values and gradients. Implementation and testing was done by my colleagues at the John von Neumann University, and they also contributed in methodological issues, developing and testing practical means of regulating accuracy and practical stopping conditions.

The objective function is the negative logarithm of an n-dimensional nondegenerate standard normal distribution function. Gradient estimates can be constructed with a reasonable effort, applying the simulation methods overviewed in Chapter 7. For the sake of simplicity, I assume that objective values are computed with a high precision. – In the present case of normally distributed random parameters, gradient computation is the bigger challenge. High-precision computation of a single non-zero component of the gradient requires an effort comparable to that of the objective value.

The proposed method is an adaptation of the randomized approach of Chapter 8. In the probabilistic context, reliability considerations are based on a special bounding approach. We demonstrate the workability of the approach with a computational experiment.

Outer approximation is the traditional solution approach in probabilistic programming. Though it is difficult to implement due to noise in gradient computation. It applies a direct cutting-plane method, and even a fairly accurate gradient may result in a cut cutting into the epigraph of the probabilistic function. One either needs sophisticated tolerance handling to avoid cutting into the epigraph, or else one needs a sophisticated convex optimization method that can handle cuts cutting into the epigraph. Yet another alternative is perpetual adjustment of existing cuts to information revealed in the course of the process.

Inner approximation of a level set results in a model that is easy to validate. The level set is approximated by means of p-efficient points. New approximation points are found by minimization over the level set. As this entails a substantial computational effort, the master part of the decomposition framework should succeed with as few p-efficient points as possible. This calls for specialized solution methods. An increasing level of complexity is noticeable in recently proposed methods.

I proposed inner approximation of the epigraph of the probabilistic function. This approach admits easier generation of new test points, and endures noise in gradient computation without any special effort. Noisy gradient estimates may yield iterates that do not improve much on our current model. But we retain a reliable inner approximation of the function. This inherent stability of the model enables the application of randomized methods of simple structure.

The proposed stochastic approximation procedure can be implemented using standard components. The master problem is conveniently solved by an off-the-shelf solver. New approximation points are found through simple line search whose direction can be determined by standard implementations of classic Monte Carlo simulation procedures.

Appendix A

Additional material

A.1 On performance profiles

Moré and associates developed means of systematic benchmarking of optimization software, see e.g., [44], [45] and the references there. Here I give a brief sketch of the concept of performance profiles, introduced by Dolan and Moré in [44]. The aim is to enable easy visual comparison of different solution methods with regard to a given set of test problems. The relative performance of each method is represented by its respective performance function $\rho : [1, +\infty) \to [0, 1]$ whose construction is sketched below.

Let $\mathcal{M}$ and $\mathcal{P}$ denote the sets of the methods and the problems, respectively. Let $P = |\mathcal{P}|$ denote the number of the problems. Let us measure the computing time $t_{p,m}$ for every method $m \in \mathcal{M}$ and problem $p \in \mathcal{P}$.

The minimal solution time

$$t_p = \min\{\, t_{p,m} \,|\, m \in \mathcal{M} \,\}$$

is used as a scaling factor in the definition of the relative performance ratio

$$r_{p,m} = \frac{t_{p,m}}{t_p}.$$

The performance function for method m is then defined as

$$\rho_m(\tau) = \frac{|\{\, p \in \mathcal{P} \,|\, r_{p,m} \leq \tau \,\}|}{P} \qquad (\tau \geq 1).$$

A.2 On linear programming primal-dual relationship

According my research into the history of the area, the equivalence of the simplex method and the dual simplex method has been known since the nineteen sixties. Prékopa in [125], Chapter 4.6 treated the simplex method and the dual simplex method in a unified form, as row and column transformations, respectively, on an appropriate square matrix. Equivalence of simplex and dual simplex steps is stated in Prékopa [131] in the following form: primal transformation formulae yield the same tableau that we obtain when we first carry out the dual transformation formulae and then take the negative transpose of the tableau.

Padberg in [120], Chapter 6.4 observes the equivalence of the primal and the dual simplex methods with the remark that the numerical behavior of the two methods may be very different.

Vanderbei in [177], Chapter 5.4 states the equivalence in the following form: there is a negative transpose relationship between the primal and the dual problem, provided that the same set of pivots that were applied to the primal problem are also applied to the dual.

Concerning the relationship between the respective bases of the primal and the dual problems, an asymmetric form is presented in Kall and Mayer [86], Chapter 1, Proposition 2.14. A symmetric form is described in [60] that can be put into words as

Observation 57 *Given a primal-dual pair of linear programming problems, there is a one-to-one mapping between the respective bases of the primal and the dual problem, which has the following characteristic. Let $\mathcal{B}$ and $\mathcal{C}$ be corresponding primal and dual bases. Then $\mathcal{C}$ is a feasible basis of the dual problem if and only if $\mathcal{B}$ is a dual feasible basis of the primal problem.*

Bibliography

[1] S. Ahmed. Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, 106:433–446, 2006.

[2] S. Ahmed. Convex relaxations of chance constrained optimization problems. *Optimization Letters*, 8:1–12, 2014.

[3] S. Ahmed, R. Garcia, N. Kong, L. Ntaimo, G. Parija, F. Qiu, and S. Sen. SIPLIB: A stochastic integer programming test problem library, 2015. http://www.isye.gatech.edu/ sahmed/siplib.

[4] K.A. Ariyawansa and A.J. Felt. On a new collection of stochastic linear programming test problems. *INFORMS Journal on Computing*, 16(3):291–299, 2004.

[5] T. Arnold, R. Henrion, A. Möller, and S. Vigerske. A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. *Pacific Journal of Optimization*, 10:5–20, 2014.

[6] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–227, 1999.

[7] E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B*, 17:173–184, 1955.

[8] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[9] J.R. Birge and F.V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34:384–392, 1988.

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

[11] J.R. Birge and R.J.-B. Wets. Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse. *Mathematical Programming Study*, 27:54–102, 1986. Special issue, A. Prékopa and R.J.-B. Wets, eds.

[12] C. Borell. Convex set functions in d-space. *Periodica Mathematica Hungarica*, 6:111–136, 1975.

[13] E. Boros and P. Veneziani. Bounds of degree 3 for the probability of the union of events. Technical report, Rutgers Center for Operations Research, 2002. RUTCOR Research Report 3-2002.

[14] H.J. Brascamp and E.H. Lieb. On extensions of the Brunn-Minkowski and Prékopa-Leindler theorems, including inequalities for log-concave functions and with an application to the diffusion equations. *Journal of Functional Analysis*, 22:366–389, 1976.

[15] J. Bukszár and A. Prékopa. Probability bounds with cherry-trees. Technical report, Rutgers Center for Operations Research, 2000. RUTCOR Research Report 44-2000.

[16] J. Bukszár and T. Szántai. Probability bounds given by hyper-cherry-trees. *Alkalmazott Matematikai Lapok*, 2:69–85, 1999. (in Hungarian).

[17] G. C. Calafiore and M. C. Campi. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46, 2005.

[18] M.C. Campi and S. Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167:155–189, 2018.

[19] A. Caré, S. Garatti, and M.C. Campi. Scenario min-max optimization and the risk of empirical costs. *SIAM Journal on Optimization*, 25:2061–2080, 2015. (winner of the 2016 stochastic programming student paper prize).

[20] A. Charnes and W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11:18–39, 1963.

[21] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.

[22] E.W. Cheney and A.A. Goldstein. Newton's method for convex programming and Tchebycheff approximation. *Numeriche Mathematik*, 1:253–268, 1959.

[23] M. Colombo and J. Gondzio. Further development of multiple centrality correctors for interior point methods. *Computational Optimization and Applications*, 41:277–305, 2008.

[24] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[25] G.B. Dantzig and A. Madansky. On the solution of two-stage linear programs under uncertainty. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 165–176. University of California Press, Berkeley, 1961.

[26] G.B. Dantzig and P. Wolfe. The decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[27] W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on-demand accuracy. *Optimization Methods and Software*, 29:1180–1209, 2014. (Published in Optimization Online in 2012).

[28] W. de Oliveira, C. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21:517–544, 2011.

[29] I. Deák. Three digit accurate multiple normal probabilities. *Numerische Mathematik*, 35:369–380, 1980.

[30] I. Deák. Computing probabilities of rectangles in case of multinormal distributions. *Journal of Statistical Computation and Simulation*, 26:101–114, 1986.

[31] I. Deák. Solving stochastic programming problems by successive regression approximations – numerical results. In K. Marti, Y. Ermoliev, and G. Pflug, editors, *Dynamic Stochastic Optimization*, volume 532 of *Lecture Notes in Economics and Mathematical Systems*, pages 209–224. Springer, 2003.

[32] I. Deák. Two-stage stochastic problems with correlated normal variables: computational experiences. *Annals of Operations Research*, 142:79–97, 2006.

[33] I. Deák. Testing successive regression approximations by large-scale two-stage problems. *Annals of Operations Research*, 186:83–99, 2011.

[34] I. Deák, H. Gassmann, and T. Szántai. Computing multivariate normal probabilities: a new look. *Journal of Statistical Computation and Simulation*, 11:920–949, 2002.

[35] I. Deák, I. Pólik, A. Prékopa, and T. Terlaky. Convex approximations in stochastic programming by semidefinite programming. *Annals of Operations Research*, 200:171–182, 2012.

[36] M.A.H. Dempster and R.R. Merkovsky. A practical geometrically convergent cutting plane algorithm. *SIAM Journal on Numerical Analysis*, 32:631–644, 1995.

[37] D. Dentcheva, B. Lai, and A. Ruszczyński. Dual methods for probabilistic optimization problems. *Mathematical Methods of Operations Research*, 60:331–346, 2004.

[38] D. Dentcheva and G. Martinez. Two-stage stochastic optimization problems with stochastic ordering constraints on the recourse. *European Journal of Operational Research*, 219:1–8, 2012.

[39] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Mathematical Programming*, 138:223–251, 2013.

[40] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89:55–77, 2000.

[41] D. Dentcheva and A. Ruszczyński. Optimization with stochastic dominance constraints. *SIAM Journal on Optimization*, 14:548–566, 2003.

[42] D. Dentcheva and A. Ruszczyński. Portfolio optimization with stochastic dominance constraints. *Journal of Banking & Finance*, 30:433–451, 2006.

[43] D. Dentcheva and A. Ruszczyński. Inverse cutting plane methods for optimization problems with second-order stochastic dominance constraints. *Optimization*, 59:323–338, 2010.

[44] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[45] E.D. Dolan, J.J. Moré, and T.S. Munson. Optimality measures for performance profiles. *SIAM Journal on Optimization*, 16:891–909, 2006.

[46] D. Drapkin, R. Gollmer, U. Gotzes, F. Neise, and R. Schultz. Risk management with stochastic dominance models in energy systems with dispersed generation. In M.I. Bertocchi, G. Consigli, and M.A.H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, volume 163 of *International Series in Operations Research & Management Science*, pages 253–271. Springer, 2011.

[47] J. Elzinga and T.G. Moore. A central cutting plane method for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[48] Y.M. Ermoliev and N.Z. Shor. Method of a random search for two-stage stochastic programming problems and its generalizations. *Kibernetica*, 1:90–92, 1968.

[49] Y.M. Ermoliev and R.J.-B. Wets, editors. *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, Berlin, 1988.

[50] C.I. Fábián. *Stochastic programming models for optical fiber manufacturing*. PhD thesis, Operációkutatási, Alkalmazott Matematikai és Statisztikai Doktori Iskola, Eötvös Loránd Tudományegyetem, 1999. Supervisor: A. Prékopa.

[51] C.I. Fábián. Bundle-type methods for inexact data. *Central European Journal of Operations Research*, 8:35–55, 2000. (Special issue, T. Csendes and T. Rapcsák, editors).

[52] C.I. Fábián. Handling CVaR objectives and constraints in two-stage stochastic models. *European Journal of Operational Research*, 191:888–911, 2008. (Special issue on Continuous Optimization in Industry, T. Illés, M. Lopez, J. Vörös, T. Terlaky, G-W. Weber, eds.) – Former version: Decomposing CVaR minimization in two-stage stochastic models. *RUTCOR Research Report* (http://rutcor.rutgers.edu) 32-2005.

[53] C.I. Fábián. Computational aspects of risk-averse optimization in two-stage stochastic models. *Optimization Online*, 2012.

[54] C.I. Fábián, E. Csizmás, R. Drenyovszki, T. Vajnai, L. Kovács, and T. Szántai. A randomized method for handling a difficult function in a convex optimization problem, motivated by probabilistic programming. *Annals of Operations Research*, 2019. DOI: 10.1007/s10479-019-03143-z. To appear in S.I.: Stochastic Modeling and Optimization, in memory of András Prékopa (editors: E. Boros, M. Katehakis, A. Ruszczyński).

[55] C.I. Fábián, E. Csizmás, R. Drenyovszki, W. van Ackooij, T. Vajnai, L. Kovács, and T. Szántai. Probability maximization by inner approximation. *Acta Polytechnica Hungarica*, 15:105–125, 2018. Special issue dedicated to the memory of András Prékopa (editors: A. Bakó, I. Maros and T. Szántai).

[56] C.I. Fábián, K. Eretnek, and O. Papp. A regularized simplex method. *Central European Journal of Operations Research*, 23:877–898, 2015.

[57] C.I. Fábián, G. Mitra, and D. Roman. Processing second-order stochastic dominance models using cutting-plane representations. *Mathematical Programming*, 130:33–57, 2011. – Former version: *Stochastic Programming E-Print Series* (www.speps.org) 10-2008.

[58] C.I. Fábián, G. Mitra, D. Roman, and V. Zverovich. An enhanced model for portfolio choice with SSD criteria: a constructive approach. *Quantitative Finance*, 11:1525–1534, 2011.

[59] C.I. Fábián, G. Mitra, D. Roman, V. Zverovich, T. Vajnai, E. Csizmás, and O. Papp. Portfolio choice models based on second-order stochastic dominance measures: an overview and a computational study. In M.I. Bertocchi, G. Consigli, and M.A.H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, volume 163 of *International Series in Operations Research & Management Science*, pages 441–469. Springer, 2011.

[60] C.I. Fábián, O. Papp, and K. Eretnek. Implementing the simplex method as a cutting-plane method, with a view to regularization. *Computational Optimization and Applications*, 56:343–368, 2013.

[61] C.I. Fábián, A. Prékopa, and O. Ruf-Fiedler. On a dual method for a specially structured linear programming problem. *Optimization Methods and Software*, 17:445–492, 2002.

[62] C.I. Fábián and Z. Szőke. Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science*, 4:313–353, 2007.

[63] C.I. Fábián and A. Veszprémi. Algorithms for handling CVaR-constraints in dynamic stochastic programming models with applications to finance. *The Journal of Risk*, 10:111–131, 2008.

[64] C.I. Fábián, C. Wolf, A. Koberstein, and L. Suhl. Risk-averse optimization in two-stage stochastic models: computational aspects and a study. *SIAM Journal on Optimization*, 25:28–52, 2015.

[65] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13:117–156, 2002.

[66] A. Frangioni. Standard bundle methods: untrusted models and duality. Technical report, Department of Informatics, University of Pisa, Italy, 2018.

[67] D.R. Fulkerson and G.B. Dantzig. Computations of maximal flows in networks. *Naval Research Logistics Quarterly*, 2:277–283, 1955.

[68] H. Gassmann. Conditional probability and conditional expectation of a random vector. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 237–254. Springer-Verlag, Berlin, 1988.

[69] H.I. Gassmann. MSLiP: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.

[70] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150, 1992.

[71] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38:284–302, 1992.

[72] J. Gondzio. HOPDM (version 2.12), a fast LP solver based on a primal-dual interior point method. *European Journal of Operational Research*, 85:221–225, 1995.

[73] J. Gondzio, P. González-Brevis, and P. Munari. Large-scale optimization with the primal-dual column generation method. *Mathematical Programming Computation*, 8:47–82, 2016.

[74] J. Hadar and W. Russel. Rules for ordering uncertain prospects. *The American Economic Review*, 59:25–34, 1969.

[75] A. Hantoute, R. Henrion, and P. Pérez-Aros. Subdifferential characterization of probability functions under Gaussian distribution. *Mathematical Programming*, 2018. DOI:10.1007/s10107-018-1237-9.

[76] G.H. Hardy, J.E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 1934.

[77] R. Henrion. Introduction to chance constraint programming. Technical report, Weierstrass-Institut für Angewandte Analysis und Stochastik, 2004. www.wias-berlin.de/people/henrion/ccp.ps.

[78] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.

[79] R. Henrion and C. Strugarek. Convexity of chance constraints with dependent random variables: the use of copulae. In M.I. Bertocchi, G. Consigli, and M.A.H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, volume 163 of *International Series in Operations Research & Management Science*, pages 427–439. Springer, 2011.

[80] J.L. Higle and S. Sen. Stochastic decomposition: an algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16:650–669, 1991.

[81] J.L. Higle and S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, volume 8 of *Nonconvex Optimization and Its Applications*. Springer, 1996.

[82] D. Holmes. A POrtable Stochastic programming Test Set (POSTS), 1995. http://users.iems.northwestern.edu/˜jrbirge/html/dholmes/post.

[83] D. Hunter. Bounds for the probability of a union. *Journal of Applied Probbility*, 13:597–603, 1976.

[84] R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. In M. Serna, R. Shaltiel, K. Jansen, and J. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 617–631. Springer, Berlin, Heidelberg, 2010. RANDOM 2010, APPROX 2010. Lecture Notes in Computer Science, vol 6302.

[85] P. Kall and J. Mayer. On testing SLP codes with SLP-IOR. In *New Trends in Mathematical Programming: Homage to Steven Vajda*, pages 115–135. Kluwer Academic Publishers, 1998.

[86] P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory, and Computation*. Springer's International Series in Operations Research and Management Science. Kluwer Academic Publishers, 2005.

[87] J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8:703–712, 1960.

[88] R. Khemchandani, A. Bhardwaj, and S. Chandra. Single asset optimal trading strategies with stochastic dominance constraints. *Annals of Operations Research*, 243:211–228, 2016.

[89] K.C. Kiwiel. An algorithm for nonsmooth convex minimization with errors. *Mathematics of Computation*, 45:171–180, 1985.

[90] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Springer-Verlag, Berlin, New York, 1985.

[91] K.C. Kiwiel. Bundle methods for convex minimization with partially inexact oracles. Technical report, Systems Research Institute, Polish Academy of Sciences, Warsaw, 2009. Available at Optimization Online.

[92] W.K. Klein Haneveld. *Duality in Stochastic Linear and Dynamic Programming*. Number 274 in Lecture Notes in Economics and Math. Systems. Springer-Verlag, 1986.

[93] W.K. Klein Haneveld and M.H. van der Vlerk. Integrated chance constraints: reduced forms and an algorithm. *Computational Management Science*, 3:245–269, 2006. – Former version: *SOM Research Report* 02A33-2002, University of Groningen.

[94] A. Koberstein, C. Lucas, C. Wolf, and D. König. Modeling and optimizing risk in the strategic gas-purchase planning problem of local distribution companies. *The Journal of Energy Markets*, 4:47–68, 2011.

[95] É. Komáromi. A dual method for probablistic constrained problems. *Mathematical Programming Study*, 28:94–112, 1986. (Special issue on stochastic programming, A. Prékopa and R.J.-B. Wets, editors).

[96] É. Komáromi. On properties of the probabilistic constrained linear programming problem and its dual. *Journal of Optimization Theory and Applications*, 55:377–390, 1987.

[97] A. Künzi-Bay and J. Mayer. Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, 3:3–27, 2006. – Former version: *Working Paper* No. 211-2005, FINRISK, Swiss National Centre of Competence in Research / Financial Valuation and Risk Management.

[98] C. Lemaréchal. Nonsmooth optimization and descent methods. Technical Report 78-4, IIASA, Laxenburg, Austria, 1978.

[99] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–147, 1995.

[100] A.Y. Levin. On an algorithm for the minimization of convex functions over convex functions. *Soviet Math. Dokl.*, 6:286–290, 1965.

[101] J. Linderoth, A. Shapiro, and S.J. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142:215–241, 2006.

[102] J.T. Linderoth and S.J. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24:207–250, 2003.

[103] X. Liu, S. Küçükyavuz, and J. Luedtke. Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*, 157:219–243, 2016.

[104] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.

[105] J. Luedtke, S. Ahmed, and G.L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122:247–272, 2010.

[106] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research and Management Science. Springer, 2008.

[107] G. Mádi-Nagy and A. Prékopa. On multivariate discrete moment problems and their applications to bounding expectations and probabilities. *Mathematics of Operations Research*, 29:229–258, 2004.

[108] J. Mayer. *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*. Gordon and Breach Science Publishers, 1998.

[109] B.L. Miller and H.M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.

[110] R.F. Muirhead. Some methods applicable to identities and inequalities of symmetric algebraic functions of n letters. *Proc. Edinburgh Math. Soc.*, 21:144–157, 1903.

[111] A. Müller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. John Wiley & Sons, Chichester, 2002.

[112] A. Nemirovski. Efficient methods in convex programming: Information-based complexity. Technion, Israel, 1994. (Lecture notes available at the author's homepage at Georgia Tech, http://www2.isye.gatech.edu/~nemirovs/).

[113] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal of Optimization*, 17:969–996, 2006.

[114] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*, volume 15 of *Wiley-Interscience Series in Discrete Mathematics*. John Wiley, New York, 1983.

[115] Yu. Nesterov and J.-Ph. Vial. Confidence level solutions for stochastic programming. *Automatica*, 44:1559–1568, 2008.

[116] N. Noyan. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers and Operations Research*, 39:541–559, 2012.

[117] W. Ogryczak. Multiple criteria linear programming model for portfolio selection. *Annals of Operations Research*, 97:143–162, 2000.

[118] W. Ogryczak and A. Ruszczyński. Dual stochastic dominance and related mean-risk models. *SIAM Journal on Optimization*, 13:60–78, 2002.

[119] W. Oliveira, C. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21:517–544, 2011.

[120] A. Padberg. *Linear optimization and extensions*. Springer, 1995.

[121] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM Journal on Computing*, 26:350–368, 1997.

[122] C. Pelekis and J. Ramon. Hoeffding's inequality for sums of weakly dependent random variables. *Mediterranean Journal of Mathematics*, 14, 2017. Article: 243.

[123] G. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In S. Uryasev, editor, *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, Norwell, MA., 2000.

[124] J. Pinter. Deterministic approximations of probability inequalities. *ZOR - Methods and Models of Operations Research*, 33:219–239, 1989.

[125] A. Prékopa. *Linear Programming*. Bolyai János Mathematical Society, Budapest, 1968. (in Hungarian).

[126] A. Prékopa. On probabilistic constrained programming. In H.W. Kuhn, editor, *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Princeton University Press, Princeton, New Jersey, 1970.

[127] A. Prékopa. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarium Mathematicarum (Szeged)*, 32:301–316, 1971.

[128] A. Prékopa. Contributions to the theory of stochastic programming. *Mathematical Programming*, 4:202–221, 1973.

[129] A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarium Mathematicarum (Szeged)*, 34:335–343, 1973.

[130] A. Prékopa. Dual method for a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution. *ZOR - Methods and Models of Operations Research*, 34:441–461, 1990.

[131] A. Prékopa. A very short introduction to linear programming. Rutgers Center for Operations Research, Rutgers University, Piscataway, NJ, 1992. RUTCOR Lecture Notes 2-1992.

[132] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, 1995.

[133] A. Prékopa, S. Ganczer, I. Deák, and K. Patyi. The STABIL stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy. In M.A.H. Dempster, editor, *Stochastic Programming*, pages 369–385. Academic Press, London, 1980.

[134] A. Prékopa and T. Szántai. Flood control reservoir system design using stochastic programming. *Mathematical Programming Study*, 9:138–151, 1978.

[135] A. Prékopa, B. Vizvári, and T. Badics. Programming under probabilistic constraint with discrete random variable. In F. Giannesi, T. Rapcsák, and S. Komlósi, editors, *New Trends in Mathematical Programming*, pages 235–255. Kluwer, Dordrecht, 1998.

[136] A. Prékopa, B. Vizvári, and G. Regős. Lower and upper bounds on probabilities of Boolean functions of events. Technical report, Rutgers Center for Operations Research, 1995. RUTCOR Research Report 36-95.

[137] J.P. Quirk and R. Saposnik. Admissibility and measurable utility functions. *Review of Economic Studies*, 29:140–146, 1962.

[138] R. Rahmaniani, T.G. Crainic, M. Gendreau, and W. Rei. Accelerating the Benders decomposition method: application to stochastic network design problems. *SIAM Journal on Optimization*, 28:875–903, 2018.

[139] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[140] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[141] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.

[142] R.T. Rockafellar. Coherent approaches to risk in optimization under uncertainty. In *Tutorials in Operations Research*, pages 38–61. INFORMS, 2007.

[143] R.T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[144] R.T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26:1443–1471, 2002.

[145] R.T. Rockafellar, S. Uryasev, and M. Zabarankin. Generalised deviations in risk analysis. *Finance and Stochastics*, 10:51–74, 2006.

[146] R. Roll. A mean/variance analysis of tracking error. *The Journal of Portfolio Management*, 18:13–22, 1992.

[147] D. Roman, K. Darby-Dowman, and G. Mitra. Portfolio construction based on stochastic dominance and target return distributions. *Mathematical Programming*, 108:541–569, 2006.

[148] D. Roman, G. Mitra, and V. Zverovich. Enhanced indexation based on second-order stochastic dominance. *European Journal of Operational Research*, 228:273–281, 2013.

[149] G. Rudolf and A. Ruszczyński. Optimization problems with second order stochastic dominance constraints: duality, compact formulations, and cut generation methods. *SIAM Journal on Optimization*, 19:1326–1343, 2008.

[150] A. Ruszczyński. A regularized decomposition method for minimizing the sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[151] A. Ruszczyński. *Nonlinear Optmization*. Princeton University Press, 2006.

[152] A. Ruszczyński and A. Świętanowski. Accelerating the regularized decomposition method for two-stage stochastic linear problems. *European Journal of Operational Research*, 101:328–342, 1997.

[153] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2:121–152, 1992.

[154] S. Sen and Y. Liu. Mitigating uncertainty via compromise decisions in two-stage stochastic linear programming: variance reduction. *Operations Research*, 64:1422–1437, 2016.

[155] A. Shapiro. Monte Carlo sampling methods. In A. Ruszczyński and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 353–425. Elsevier, Amsterdam, 2003.

[156] A. Shapiro and T. Homem de Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81:301–325, 1998.

[157] Yongjia Song and J. Luedtke. An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. *SIAM Journal on Optimization*, 25:1344–1367, 2015.

[158] G. Sonnevend. An "analytical centre" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prékopa, J. Szelezsán, and B. Strazicky, editors, *System Modelling and Optimization*, pages 866–875. Springer, Berlin, Heidelberg, 1986. (Volume 84 of the Lecture Notes in Control and Information Sciences.

[159] G. Sonnevend. New algorithms in convex programming based on a notion of 'centre' (for systems of analytic inequalities) and on rational expectations. In K.H. Hoffmann, J.-B. Hirriart-Urruty, C. Lemaréchal, and J. Zowe, editors, *Trends in Mathematical Optimization: Proceedings of the 4th French-German Conference on Optimization*, pages 311–327. Birkhäuser, Basel, Switzerland, 1988.

[160] Hailin Sun, Huifu Xu, R. Meskarian, and Yong Wang. Exact penalization, level function method, and modified cutting-plane method for stochastic programs with second order stochastic dominance constraints. *SIAM Journal on Optimization*, 23:602–631, 2013.

[161] T. Szántai. A procedure for determination of the multivariate normal probability distribution function and its gradient values. *Alkalmazott Matematikai Lapok*, 2:27–39, 1976. (in Hungarian).

[162] T. Szántai. *Numerical Evaluation of Probabilities Concerning Multidimensional Probability Distributions, Thesis.* Hungarian Academy of Sciences, Budapest, 1985.

[163] T. Szántai. A computer code for solution of probabilistic-constrained stochastic programming problems. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 229–235. Springer-Verlag, Berlin, 1988.

[164] T. Szántai. Probabilistic constrained programming and distributions with given marginals. In J. Stepan V. Benes, editor, *Distributions with Given Marginals and Moment Problems*, pages 205–210, 1997.

[165] T. Szántai. Improved bounds and simulation procedures on the value of the multivariate normal probability distribution function. *Annals of Operations Research*, 100:85–101, 2000.

[166] P. Szász. *Elements of Differential and Integral Calculus (in Hungarian).* Közoktatásügyi Kiadóvállalat, Budapest, 1951.

[167] Y. Takano, K. Nanjo, N. Sukegawa, and S. Mizuno. Cutting plane algorithms for mean-CVaR portfolio optimization with nonconvex transaction costs. *Computational Management Science*, 12:319–340, 2015.

[168] E. Tamm. On g-concave functions and probability measures (in Russian). *Eesti NSV Teaduste Akademia Toimetised, Füüsika-Matemaatika (News of the Estonian Academi of Sciences)*, 26:376–379, 1977.

[169] I. Tomescu. Hypertrees and Bonferroni inequalities. *Journal of Combinatorial Theory, Series B*, 41:209–217, 1986.

[170] S. Trukhanov, L. Ntaimo, and A. Schaefer. Adaptive multicut aggregation for two-stage stochastic linear programs. *European Journal of Operational Research*, 206:395–406, 2010.

[171] P. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73:291–341, 1996.

[172] C.A. Valle, D. Roman, and G. Mitra. Novel approaches for portfolio construction using second order stochastic dominance. *Computational Management Science*, 14:257–280, 2017.

[173] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[174] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computational Optimization and Applications*, 57:555–597, 2014.

[175] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24:733–765, 2014.

[176] R. van Slyke and R.J.-B. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[177] R.J. Vanderbei. *Linear Programming. Foundations and Extensions*. International Series in Operations Research & Management Science. Springer, 2001 and 2008.

[178] A.F. Veinott. The supporting hyperplane method for unimodal programming. *Operations Research*, 15:147–152, 1967.

[179] M.T. Vespucci, M. Bertocchi, L. Escudero, M. Innorta, and S. Zigrino. A stochastic model for generation expansion planning in the long period with different risk measures. In L. Suhl, G. Mitra, C. Lucas, A. Koberstein, and L. Beckmann, editors, *Applied Mathematical Optimization and Modelling. Extended Abstracts of the APMOD 2012 Conference*, volume 8 of *DSOR Contributions to Information Sytems*, pages 114–121. DS&OR Lab, University of Paderborn, 2012.

[180] R.J.-B. Wets. Stochastic programs with fixed recourse: The equivalent deterministic program. *SIAM Review*, 16:309–339, 1974.

[181] G.A. Whitmore and M.C. Findlay. *Stochastic Dominance: An Approach to Decision- Making Under Risk*. D.C.Heath, Lexington, MA, 1978.

[182] P.G. Wodehouse. *Something New*. D. Appleton & Co., New York City, U.S.A., 1915. Available at http://www.classicreader.com/.

[183] C. Wolf. *Advanced acceleration techniques for nested Benders decomposition in stochastic programming*. PhD thesis, University of Paderborn, 2013. (Available at https://d-nb.info/1046905090/34) Advisors L.Suhl and C.I. Fábián.

[184] C. Wolf, C.I. Fábián, A. Koberstein, and L. Suhl. Applying oracles of on-demand accuracy in two-stage stochastic programming - a computational study. *European Journal of Operational Research*, 239:437–448, 2014.

[185] C. Wolf and A. Koberstein. Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method. *European Journal of Operational Research*, 230:143–156, 2013.

[186] K.J. Worsley. An improved Bonferroni inequality and applications. *Biometrika*, 69:297–302, 1982.

[187] G. Zakeri, A.B. Philpott, and D.M. Ryan. Inexact cuts in Benders decomposition. *SIAM Journal on Optimization*, 10:643–657, 2000.

[188] G. Zoutendijk. *Methods of Feasible Directions: A Study in Linear and Non-Linear Programming*. Elsevier Publishing Co., Amsterdam, 1960.

[189] V. Zverovich. *Modelling and solution methods for stochastic optimisation*. LAP LAMBERT Academic Publishing, Saarbrücken, 2012. (PhD thesis. School of Information Systems, Computing and Mathematics, Brunel University of West London, 2011. Advisor: G. Mitra).

[190] V. Zverovich, C.I. Fábián, E.F.D. Ellison, and G. Mitra. A computational study of a solver system for processing two-stage stochastic LPs with enhanced benders decomposition. *Mathematical Programming Computation*, 4:211–238, 2012.

[191] V. Zverovich, C. Arbex Valle, F. Ellison, and G. Mitra. *FortSP: A Stochastic Programming Solver. Manual.* OptiRisk Systems, 2014. Available at http://optirisk-systems.com/manuals/FortspManual.pdf.