

Solving two-stage stochastic programming problems with level decomposition

Csaba I. Fábián · Zoltán Szóke

Published online: 22 August 2006
© Springer-Verlag 2006

Abstract We propose a new variant of the two-stage recourse model. It can be used e.g., in managing resources in whose supply random interruptions may occur. Oil and natural gas are examples for such resources. Constraints in the resulting stochastic programming problems can be regarded as generalizations of integrated chance constraints. For the solution of such problems, we propose a new decomposition method that integrates a bundle-type convex programming method with the classic distribution approximation schemes. Feasibility and optimality issues are taken into consideration simultaneously, since we use a convex programming method suited for constrained optimization. This approach can also be applied to traditional two-stage problems whose recourse functions can be extended to the whole space in a computationally efficient way. Network recourse problems are an example for such problems. We report encouraging test results with the new method.

Keywords Stochastic recourse models · Decomposition · Successive approximation

Mathematics Subject Classification 90C15 Stochastic programming

C. I. Fábián (✉)
Department of Operations Research, Loránd Eötvös University of Budapest, P.O. Box 120,
Budapest 1518, Hungary
e-mail: fabian@cs.elte.hu

Z. Szóke
Doctoral School in Applied Mathematics, Loránd Eötvös University of Budapest, Budapest,
Hungary

0 Introduction

In this paper, we identify classes of two-stage recourse problems where the expected recourse function can be extended to the whole space in a computationally efficient way. Network recourse problems constitute such a problem class. Two-stage recourse problems belonging to such classes can be formulated as constrained convex programming problems. This formulation is called *standard model* in the paper.

We also propose a new variant of the two-stage recourse model. Called *relaxed model* in the paper, this can be regarded as a generalization of the well-known *integrated chance constrained* model of Klein Haneveld (1986). We also give economic interpretations of the relaxed model. One of them substantially differs from traditional two-stage recourse models. It deals with the design and operation of a system that consumes certain resources in whose supply random interruptions may occur.

For the above models, we propose a new solution method that we call Level Decomposition. The method consists of two known components:

- The first component is an approximative version of the Constrained Level Method of Lemaréchal et al. (1995). The approximative method was proposed by Fábián (2000).
- The second component involves the classic discretization and bounding methods worked out by Kall (1980), Kall and Stoyan (1982), and Birge and Wets (1986).

The novelty of Level Decomposition is the way these two components are integrated, and the fact that the master problem is solved by a convex programming method suited for constrained optimization.

In a typical implementation of the traditional solution framework, the main module is the distribution approximation scheme, and the submodule is the two-stage stochastic problem solver. To what extent the approximation should be refined between solver calls, is a difficult question, to which Mayer (1998) proposed a heuristic method. In Level Decomposition, the main module is the solver, and the approximating coarse distributions are refined in accordance with the solution process of the master problem.

Using a convex programming method suited for constrained optimization means that feasibility and optimality issues are taken into consideration simultaneously. The bundle-type regularization extends to both feasibility and optimality issues. We avoid feasibility cuts that are constituents of the Benders family of solution methods.

We report encouraging test results with Level Decomposition.

The paper is organized as follows

In the remaining part of this section we outline the traditional two-stage recourse model, and the classic decomposition methods used for its solution. Section 1 formulates two-stage recourse models and new variants.

Section 2 describes the inexact convex programming method to be used in the decomposition. Section 3 gives an algorithmic description of Level Decomposition.

Section 4 deals with implementation issues. Test problems are described and parameter settings specified in Section 5. Test results are reported in section 6.

The results of the paper are summarized and potential fields of application are suggested in section 7.

The models and the solution method proposed in this paper were worked out by the first author. The method was implemented and tested with joint effort of the authors.

0.1 The traditional two-stage recourse model and the classic decomposition methods

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 may be to find a balance between investment cost and long-term operation costs. Comprehensive treatment of these models and their solution methods can be found in Kall and Wallace (1994), Prékopa (1995), Birge and Louveaux (1997), Mayer (1998), Ruszczyński and Shapiro (2003), and Kall and Mayer (2005).

In case the random parameters do not have a finite distribution, discretization methods are applied to make the problem tractable for solvers. Description of these methods can be found in the above-mentioned books, a practical and comprehensive one in Mayer (1998). Although these methods were originally devised for the discrete approximation of a continuous distribution, they can as well be applied to approximate a discrete distribution with a coarser one. Mayer successfully applied discretization methods for this purpose. In the context of discrete finite distributions, the possible outcomes of the random events are called *scenarios*.

In this paper we consider linear models. There are two traditional ways of formulating the problem as a mathematical programming problem:

- (a) as a linear programming problem of a specific structure: For each scenario, a subproblem is included that describes the second-stage decision in case this scenario realizes. These subproblems are called *recourse problems*. The subproblems are linked by the *first-stage variables*, i.e., the variables representing the first-stage decision.
- (b) as a convex programming problem on the first-stage variables. Recourse problems are not explicitly included, but feasibility of the recourse problems is enforced by cuts on the feasible region. The objective function is not known explicitly, but supporting hyperplanes can be constructed to its graph.

The above two formulations are equivalent from a mathematical point of view. They only differ in that they suggest (apparently) different solution approaches. Dantzig and Madansky (1961) observed that the dual of the linear programming problem (a) fits the prototype for the Dantzig–Wolfe decomposition (Dantzig and Wolfe 1960). Based on the convex programming (b) formulation, Van Slyke and Wets (1969) worked out a cutting-plane method called L-shaped Method. It turned out to be the same as the Benders (1962) decomposition method, specialized for the linear programming problem (a). Hence the L-shaped Method on the one hand, and the Dantzig–Wolfe decomposition applied to the dual of the problem (a) on the other hand, are identical methods. The cutting-plane formulation has the advantage that it gives a clear visual impression of the procedure.

In their pure forms, decomposition or cutting-plane methods are impractical: they are known to be unstable, the convergence is slow, and hence the number of the master iterations is large. Implementations must deal with a constantly growing master problem. (There is no effective constraint reduction strategy, because the master problem tends to be dual degenerate.) Several stochastic programming decomposition methods were proposed to improve efficiency. Practical descriptions can be found in Mayer (1998), Ruszczyński (2003), and in Kall and Mayer (2005).

The Regularized Decomposition Method of Ruszczyński (1986) successfully mitigates the above-mentioned difficulties. It is a bundle-type method for the minimization of the sum of polyhedral convex functions over a convex polyhedron. The bundle-type penalty term in the objective function of the master problem stabilizes the procedure, and decreases the number of the iterations required. Moreover, this method lays an emphasis on keeping the master problem of a tractable size. The penalty term in the objective function eliminates dual degeneracy, and enables an effective constraint reduction strategy.

In the Regularized Decomposition Method, feasibility of the solution is attained by imposing explicit constraints on the first-stage variables. These constraints, called *feasibility cuts*, are constituents of the Benders family of decomposition methods. Imposing feasibility cuts may cause the scope of optimization to alternate between minimizing the objective function and finding a solution that satisfies existing cuts.

Implementation strategies of the Regularized Decomposition (RD) Method for two-stage stochastic problems are described by Ruszczyński and Świętowski (1997). They implemented the method, and solved several problems, each with a growing scenario set. Their test results show that the number of the RD master iterations required is a slowly increasing function of the number of the scenarios.

Inexact cuts represent another possibility of improvement on pure-form decomposition methods. Inexact cuts were used by Szántai (1988) in a heuristic scheme for the solution of probabilistic constrained stochastic programming problems.

For two-stage stochastic programming problems with *relatively complete recourse*, an inexact cut method was proposed by Zakeri et al. (2000).

(Relatively complete recourse means that no extra action is needed to enforce feasibility of the recourse problems.) Before starting the procedure, they choose a decreasing sequence of tolerances which converges to 0. In the i -th iteration, the inexact cut is generated with a tolerance not larger than the i -th element in the pre-set sequence. Zakeri et al. (2000) report substantial improvements over the exact (pure) decomposition method. They did not apply any distribution approximation scheme. Accuracy of the cuts was regulated through decreasing the stopping tolerance for the linear programming solver used for solving the recourse problems.

1 New variants of two-stage recourse models

The first-stage and second-stage decision vectors will be denoted by $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^s$, respectively. The random event will be represented by the random vector $\boldsymbol{\xi}$. Realizations of $\boldsymbol{\xi}$ will be denoted by $\boldsymbol{\xi}_\omega$ ($\omega \in \Omega$). Formally, (Ω, \mathcal{A}, P) is a probability space. The probability distribution can be either continuous or discrete.

There are two optimization problems to be solved. The *second-stage problem* or recourse problem is formulated by assuming that the first-stage decision has been made, and the random event has been observed. Supposing that the realization $\boldsymbol{\xi}_\omega$ has occurred ($\omega \in \Omega$), the recourse problem is

$$\begin{aligned} & \min \mathbf{q}^T \mathbf{y} \\ & \text{subject to} \\ & T_\omega \mathbf{x} + W \mathbf{y} = \mathbf{d}_\omega, \\ & \mathbf{y} \geq \mathbf{0}, \end{aligned} \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^s$, $\mathbf{d}_\omega \in \mathbb{R}^r$ are given vectors, and $T_\omega \in \mathbb{R}^{r \times n}$, $W \in \mathbb{R}^{r \times s}$ are given matrices. Formally, \mathbf{d} is a random vector, T is a random matrix; and the random vector $\boldsymbol{\xi}$ consists of the components of \mathbf{d} and T .

Having fixed $\omega \in \Omega$, let K_ω denote the set of those \mathbf{x} vectors for which the recourse problem (1) is feasible. This is a convex polyhedron that we assume to be nonempty. For $\mathbf{x} \in K_\omega$, let $f(\mathbf{x}, \boldsymbol{\xi}_\omega)$ denote the optimal objective value of the recourse problem (1). We assume that $f(\mathbf{x}, \boldsymbol{\xi}_\omega) > -\infty$ ($\mathbf{x} \in K_\omega$). Or equivalently, we assume that the dual of the recourse problem is feasible:

$$\begin{aligned} & \max \mathbf{z}^T (\mathbf{d}_\omega - T_\omega \mathbf{x}) \\ & \text{subject to} \\ & \mathbf{z}^T W \leq \mathbf{q}^T. \end{aligned} \tag{2}$$

Feasibility of the dual problem does not depend on \mathbf{x} or $\boldsymbol{\xi}$. The function $f(\cdot, \boldsymbol{\xi}_\omega) : K_\omega \rightarrow \mathbb{R}$ is a *polyhedral* (i.e., piecewise linear) convex function.

Customary formulation prescribes that the second-stage problems should be feasible for any realization of $\boldsymbol{\xi}$. Formally, \mathbf{x} should belong to the convex set $K := \bigcap_{\omega \in \Omega} K_\omega$. If the distribution is discrete finite then K is a polyhedron.

The customary formulation of the *first-stage problem* that represents the first decision, is as follows:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + \mathbb{E}(f(\mathbf{x}, \boldsymbol{\xi})) \\ \text{subject to} \quad & \\ & A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \\ & \mathbf{x} \in K, \end{aligned} \tag{3}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ are given vectors, and $A \in \mathbb{R}^{m \times n}$ is a given matrix. This model is called a *fixed recourse* model, since the matrix W is not random. Though a random \mathbf{q} vector is also allowed generally. The model originates from Dantzig (1955) and Beale (1955), and mathematical characterization was given by Wets (1974).

In the present special case it is easy to see that the expectation in the objective exists and is finite, provided $\boldsymbol{\xi}$ has a finite expectation, which we assume henceforth. $F(\mathbf{x}) := \mathbb{E}(f(\mathbf{x}, \boldsymbol{\xi}))$ is called *expected recourse function*. This is a convex function with the domain K .

In this paper we assume that the polyhedron $X := \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is nonempty and bounded. Let us introduce auxiliary variables in the recourse problems (1). The new variables are represented by the vector $\mathbf{s} = (s_1, \dots, s_r)^T$. We add a new term to the objective function which penalizes infeasibility. Here w is a fixed (large) weight, and $\|\mathbf{s}\|_\diamond$ is a norm of the auxiliary vector:

$$\begin{aligned} \min \quad & \mathbf{q}^T \mathbf{y} + w\|\mathbf{s}\|_\diamond \\ \text{subject to} \quad & \\ & T_\omega \mathbf{x} + W\mathbf{y} + \mathbf{s} = \mathbf{d}_\omega, \\ & \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{4}$$

In order retain linearity, we use either $\|\mathbf{s}\|_\diamond = \|\mathbf{s}\|_1 := |s_1| + \dots + |s_r|$ or $\|\mathbf{s}\|_\diamond = \|\mathbf{s}\|_{\max} := \max\{|s_1|, \dots, |s_r|\}$.

The above problem is a generalization of the formulation proposed by Prékopa (1995, chapter 12.10). Having formulated the recourse problem with inequality constraints, Prékopa introduced non-negative slack variables, and penalized them with different positive weights. Equation (4) is indeed a generalization: the equality constraints in (4) can be transformed into inequalities by including negative unit vectors in W ; and different penalties for different slack variables can be expressed by row scaling. Prékopa observes that we get an extension of the recourse function if the penalties are large enough, but he does not examine the necessary magnitude. In the forthcoming *standard model* formulation we will describe a sufficient condition.

Via equivalent linear programming formulations, it is easy to verify that the dual of problem (4) takes the form

$$\begin{aligned} \max \quad & \mathbf{z}^T (\mathbf{d}_\omega - T_\omega \mathbf{x}) \\ \text{subject to} \quad & \\ & \mathbf{z}^T W \leq \mathbf{q}^T, \\ & \|\mathbf{z}\|_\square \leq w, \end{aligned} \tag{5}$$

where the norm $\| \cdot \|_{\square}$ depends on the selection of the norm $\| \cdot \|_{\diamond}$ in (4): for $\| \cdot \|_{\diamond} = \| \cdot \|_1$ we have $\| \cdot \|_{\square} = \| \cdot \|_{\max}$; and for $\| \cdot \|_{\diamond} = \| \cdot \|_{\max}$ we have $\| \cdot \|_{\square} = \| \cdot \|_1$. In the forthcoming discussion, we will consider problems (4) and (5) as linear programming problems.

Problem (4) is feasible for any $\mathbf{x} \in \mathbb{R}^n$. Let $f_w(\mathbf{x}, \xi_{\omega})$ denote the optimal objective value (possibly $-\infty$). This is increasing in w . We show that the objective function is bounded if w is large enough: Let \mathbf{z} denote a feasible solution of (2) that we have assumed to exist. Let $w := \|\mathbf{z}\|_{\square}$. If we have $w \geq \underline{w}$ then \mathbf{z} is a feasible solution of (5), providing a lower bound for the objective value of (4). The bound \underline{w} is easily computable and has a reasonable magnitude. In what follows we assume that $w \geq \underline{w}$ holds. Hence we have $f_w(\mathbf{x}, \xi_{\omega}) > -\infty$ for any $\mathbf{x} \in \mathbb{R}^n$.

$f_w(\cdot, \xi_{\omega})$ is a polyhedral convex function, and $f_w(\mathbf{x}, \xi_{\omega}) \leq f(\mathbf{x}, \xi_{\omega})$ holds for any $\mathbf{x} \in K_{\omega}$. Hence the expectation $F_w(\mathbf{x}) := \mathbb{IE}(f_w(\mathbf{x}, \xi))$ is a convex function, and $F_w(\mathbf{x}) \leq F(\mathbf{x})$ holds for any $\mathbf{x} \in K$. The constraint $\|\mathbf{z}\|_{\square} \leq w$ in (5) implies that the function $f_w(\cdot, \xi_{\omega})$ is Lipschitz continuous with the constant $Cw\|T_{\omega}\|$. (The multiplier C depends only on the norm used and the dimension r .) Hence the expectation function $F_w(\mathbf{x})$ is also Lipschitz continuous with the constant $Cw \mathbb{IE}(\|T\|)$. Finiteness of $\mathbb{IE}(\|T\|)$ follows from ξ having a finite expectation.

Let $g(\mathbf{x}, \xi_{\omega})$ denote the measure of the inconsistency of the recourse problem, defined as follows:

$$\begin{aligned}
 g(\mathbf{x}, \xi_{\omega}) &:= \min \|\mathbf{s}\|_{\diamond} \\
 &\text{subject to} \\
 T_{\omega}\mathbf{x} + W\mathbf{y} + \mathbf{s} &= \mathbf{d}_{\omega}, \\
 \mathbf{y} &\geq \mathbf{0}.
 \end{aligned}
 \tag{6}$$

This problem has an optimal solution for any $\mathbf{x} \in \mathbb{R}^n$. $g(\cdot, \xi_{\omega})$ is a non-negative polyhedral convex function, and we have $g(\mathbf{x}, \xi_{\omega}) = 0$ if and only if $\mathbf{x} \in K_{\omega}$. By considering the dual of (6) it is easy to see that $g(\cdot, \xi_{\omega})$ is Lipschitz continuous with the constant $C\|T_{\omega}\|$. Let $G(\mathbf{x}) := \mathbb{IE}(g(\mathbf{x}, \xi))$ denote the expectation of the inconsistency measure. Our assumptions assure finiteness of the expectation. $G(\mathbf{x})$ is a convex function, and Lipschitz continuous with the constant $C \mathbb{IE}(\|T\|)$. Owing to the non-negativity of $g(\mathbf{x}, \xi_{\omega})$, we have that $\mathbf{x} \in K$ is equivalent to $G(\mathbf{x}) = 0$. Instead of this equation we will use the further equivalent constraint $G(\mathbf{x}) \leq 0$ that is more convenient in convex programming context.

We will consider two convex programming models.

1.1 Standard model

This is but a re-statement of the traditional two-stage linear stochastic programming problem, which is only possible under the following assumption:

Assumption 1 The function $f_w(\cdot, \xi_{\omega})$ is an extension of $f(\cdot, \xi_{\omega})$, i.e. $f_w(\mathbf{x}, \xi_{\omega}) = f(\mathbf{x}, \xi_{\omega})$ ($\mathbf{x} \in K_{\omega}$) holds for each $\omega \in \Omega$.

From Assumption 1 it follows that the function F_w is an extension of the expected recourse function F . In this model, we will denote the extended functions by $f(\cdot, \xi_\omega)$ and F , omitting subscripts. With this notation, the first-stage problem (3) can be written in the following form:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + F(\mathbf{x}) \\ \text{subject to} \quad & \\ & \mathbf{x} \in X, \\ & G(\mathbf{x}) \leq 0. \end{aligned} \tag{7}$$

In this model we assume that $K \cap X$ is not empty.

It is easy to verify that Assumption 1 holds for large enough w : Let

$$\bar{w} := \max \|\mathbf{z}\|_\square$$

while \mathbf{z} sweeps through the set of the feasible basic solutions of the system $\mathbf{z}^T W \leq \mathbf{q}^T$, i.e., of the constraint set of problem (2). (Basic solutions are defined through first converting the system into an equality system by adding slack variables.)

Proposition 2 *Assumption 1 holds if we have $w \geq \bar{w}$.*

Proof Given $\omega \in \Omega$ and $\mathbf{x} \in K_\omega$, the problem (2) has an optimal solution. Hence an optimal basic solution also exists, let \mathbf{z}^* denote one of those.

Since we have $\|\mathbf{z}^*\|_\square \leq \bar{w}$ by the definition of \bar{w} , it follows that \mathbf{z}^* is a feasible solution of (5) for $w \geq \bar{w}$. Problem (5) being a tightened version of (2), \mathbf{z}^* is also an optimal solution of (5). □

\bar{w} is not easily computed, but there are well-known upper bounds for the norm of a basic feasible solution of a linear programming problem, see e.g. Schrijver (1986). In general, these bounds are too large for practical applications. Though in many special cases of practical interest, usable bounds can be computed for \bar{w} :

- Combinatorial nature of the recourse problems can often be exploited. For example, if we have a network recourse problem, then forthcoming Proposition 3 gives a usable bound.
- If W is an integer matrix, then the bound given by Klafszky and Terlaky (1992) will often be of a usable magnitude.
- If the matrix W contains only non-negative components, then a bound can be constructed using the minimal nonzero element.

Proposition 3 *If problem (1) is a network problem, and the norm $\|\cdot\|_\diamond = \|\cdot\|_1$ is used, then we have $\bar{w} \leq \|\mathbf{q}\|_1$. In case the norm $\|\cdot\|_\diamond = \|\cdot\|_{\max}$ is used, we have $\bar{w} \leq r\|\mathbf{q}\|_1$.*

Proof Easily follows from the construction of the dual vector in the network simplex method, see e.g. Kennington and Helgason (1980). □

Remark 4 Assumption 1 does not generally make the constraint $G(\mathbf{x}) \leq 0$ redundant in problem (7). Although in case of a discrete finite probability distribution, this constraint indeed becomes redundant for large enough w , this magnitude may be excessively large. Typically this is the case if we have a large number of scenarios with small individual probabilities.

Remark 5 When we start solving a general problem, we may not know whether the weight w has been set large enough. However, it is enough to check Assumption 1 in an optimal solution:

Suppose that having set the penalty w by guess, we solved the problem

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + F_w(\mathbf{x}) \\ \text{subject to} \quad & \\ & \mathbf{x} \in X, \\ & G(\mathbf{x}) \leq 0. \end{aligned}$$

Let \mathbf{x}^* denote an optimal solution (entailing $\mathbf{x}^* \in K$).

Assume that $f_w(\mathbf{x}^*, \xi_\omega) = f(\mathbf{x}^*, \xi_\omega)$ holds for each $\omega \in \Omega$. It follows that $F_w(\mathbf{x}^*) = F(\mathbf{x}^*)$. Since we have $F_w(\mathbf{x}) \leq F(\mathbf{x})$ ($\mathbf{x} \in K$), it follows that \mathbf{x}^* is an optimal solution of the original problem (3).

Checking Assumption 1 in a certain point causes no theoretical difficulty if the distribution is finite discrete.

1.2 Relaxed model

The following convex programming problem is a relaxation of the traditional two-stage linear stochastic programming problem:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} + F_w(\mathbf{x}) \\ \text{subject to} \quad & \\ & \mathbf{x} \in X, \\ & G(\mathbf{x}) \leq \Gamma, \end{aligned} \tag{8}$$

where $\Gamma \geq 0$ is a given parameter (assumed to be large enough to make the problem feasible).

Interpretation of the relaxed model depends on whether the following assumption holds:

Assumption 6 For each $\omega \in \Omega$ and $\mathbf{x} \in \mathbb{R}^n$, an optimal basic solution $(\mathbf{y}^*, \mathbf{s}^*)$ of problem (4) is also an optimal solution of problem (6).

It is easy to verify that Assumption 6 holds for large enough w : Given $\omega \in \Omega$, let us consider the affine half-space

$$H := \{\mathbf{x} \in \mathbb{R}^n\} \times \{w \geq \underline{w}\} \subset \mathbb{R}^{n+1}.$$

(The bound \underline{w} had been defined earlier to guarantee that problem (4) has an optimal solution.) Given a basis \mathcal{B} of problem (4), let $\mathcal{F}_{\mathcal{B}}$ denote the set of those \mathbf{x} vectors for which \mathcal{B} is a feasible basis. $\mathcal{F}_{\mathcal{B}}$ is a closed convex polyhedron. Similarly, let $\mathcal{D}_{\mathcal{B}}$ denote the set of those $w \geq \underline{w}$ values for which \mathcal{B} is a dual feasible basis. $\mathcal{D}_{\mathcal{B}}$ is a closed interval (possibly infinite). \mathcal{B} is an optimal basis for $(\mathbf{x}, w) \in \mathcal{F}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{B}}$. Problem (4) has an optimal solution, hence an optimal basis exists, for any $\mathbf{x} \in \mathbb{R}^n$ and $w \geq \underline{w}$. It follows that taking all bases into consideration, the prism-like sets $\mathcal{F}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{B}}$ cover the affine half-space H .

Since there are finitely many bases, there exists a bound \overline{w} such that the ray $\{w \geq \overline{w}\}$ is disjoint from each finite interval $\mathcal{D}_{\mathcal{B}}$. The affine half-space

$$\{\mathbf{x} \in \mathbb{R}^n\} \times \{w \geq \overline{w}\} \subseteq H$$

is covered exclusively by such prism-like sets $\mathcal{F}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{B}}$ for which the interval $\mathcal{D}_{\mathcal{B}}$ is infinite.

Though the set $\mathcal{F}_{\mathcal{B}}$ depends on ω , the set $\mathcal{D}_{\mathcal{B}}$ does not. It follows that the bound \overline{w} is the same for each $\omega \in \Omega$.

Proposition 7 *Assumption 6 holds if we have $w \geq \overline{w}$.*

Proof Given $\omega \in \Omega$ and $\mathbf{x} \in \mathbb{R}^n$, let \mathcal{B}^* be such basis that $\mathcal{F}_{\mathcal{B}^*} \times \mathcal{D}_{\mathcal{B}^*}$ contains the point $(\mathbf{x}, \overline{w})$. Let $(\mathbf{y}^*, \mathbf{s}^*)$ denote the basic solution belonging to \mathcal{B}^* .

Due to the selection of \overline{w} , the prism-like set $\mathcal{F}_{\mathcal{B}^*} \times \mathcal{D}_{\mathcal{B}^*}$ contains the whole ray $\{\mathbf{x}\} \times \{w \geq \overline{w}\}$. Hence $(\mathbf{y}^*, \mathbf{s}^*)$ is an optimal solution of (4) for any $w \geq \overline{w}$.

We show that $(\mathbf{y}^*, \mathbf{s}^*)$ is also an optimal solution of problem (6). Indeed, problems (4) and (6) have identical feasible sets. Let (\mathbf{y}, \mathbf{s}) be a feasible solution. Then

$$\mathbf{q}^T \mathbf{y} - \mathbf{q}^T \mathbf{y}^* \geq w (\|\mathbf{s}^*\|_{\diamond} - \|\mathbf{s}\|_{\diamond})$$

holds for any $w \geq \overline{w}$. The left-hand side does not depend on w , hence the right-hand side needs to be non-positive. □

Although Assumption 6 is in general stronger than Assumption 1, we have

Proposition 8 *Any number $w, w > \overline{w}$ is an appropriate selection for the bound \overline{w} .*

Proof The problems (4) and (5) are a primal-dual pair of linear programming problems. For any such pair of problems, there is a natural one-to-one mapping between the set of the bases of the primal problem, and the set of the bases of the dual problem. We sketch the construction of this mapping.

In both problems, let us add a respective slack variable to each constraint. The feasible interval of a slack variable is determined so that the corresponding constraint will transform into equality. For a constraint originally in equality form, we add a slack variable whose feasible interval consist of the single value 0. (A problem with slacks added this way is said to be in *computational form*

by Maros (2003b).) The column indexes originally existing in the problems are called *structurals* as opposed to slacks.

There is a natural matching between primal slack indexes and dual structural indexes. Similarly, there is a natural matching between primal structural indexes and dual slack indexes.

Let \mathcal{B} be a basis of the primal problem. Its natural pair \mathcal{B}' will be identified by selecting a certain set of column indexes in the dual problem. Namely, \mathcal{B}' will contain those dual structural indexes that match primal slack indexes not basic in \mathcal{B} . Similarly, \mathcal{B}' will contain those dual slack indexes that match primal structural indexes not basic in \mathcal{B} .

\mathcal{B}' is indeed a basis of the dual problem: Let B and B' denote the matrices corresponding to \mathcal{B} and \mathcal{B}' , respectively (i.e., the square matrix B contains those columns of the primal problem that are basic in \mathcal{B} . The square matrix B' contains those columns of the dual problem that belong to \mathcal{B}') The common part of B and B' is a square submatrix that we denote by S . To be precise, S is a submatrix of B , and S^T is a submatrix of B' . The column indexes belonging to S are the primal structural indexes that are basic in \mathcal{B} . The row indexes belonging to S in the primal problem are those whose respective slack variables are not basic in \mathcal{B} . Since B is not singular, it follows that S is not singular. From the latter, it follows that B' is not singular.

It is easily seen that \mathcal{B}' is a feasible basis of the dual problem if and only if \mathcal{B} is a dual feasible basis of the primal problem.

In the present case, \bar{w} was defined in such a manner that problem (5) has the same set of feasible bases for any $w > \bar{w}$. (If the feasible domain of problem (2) is unbounded, then the basic solutions belonging to the same feasible basis \mathcal{B}' for different $w > \bar{w}$ values, are represented by different points on the same ray of the feasible domain.)

From the natural mapping of the bases, it follows that problem (4) has the same set of dual feasible bases for any $w > \bar{w}$. Hence the ray $\{w > \bar{w}\}$ is disjunct from each finite interval $\mathcal{D}_{\mathcal{B}}$. \square

Corollary 9 *In case of network recourse problems, the bound given in Proposition 3 assures that Assumption 6 also holds.*

Namely, any number w , $w > \|q\|_1$ is an appropriate selection for \bar{w} , in case the norm $\|\cdot\|_{\diamond} = \|\cdot\|_1$ is used. And any number w , $w > r\|q\|_1$ is an appropriate selection in case the norm $\|\cdot\|_{\diamond} = \|\cdot\|_{\max}$ is used.

In the remaining part of this section we show the relationship between the relaxed model and the well-known and frequently used integrated chance constraints. Moreover we present interpretations of the relaxed model.

Relationship with integrated chance constraints. Assume that we have $W = -I$, $q = \mathbf{0}$.

Having fixed $\omega \in \Omega$, the recourse problem (1) is feasible if and only if $T_{\omega}x \geq d_{\omega}$ holds (e.g., in a financial application, $T_{\omega}x$ could stand for the random yield of a portfolio x which should at least meet a certain benchmark d).

Let us use the norm $\| \cdot \|_{\diamond} = \| \cdot \|_{\max}$. The inconsistency measure $g(\mathbf{x}, \boldsymbol{\xi}_{\omega})$ defined in (6) reduces to

$$g(\mathbf{x}, \boldsymbol{\xi}_{\omega}) := \left\| \left[\mathbf{d}_{\omega} - T_{\omega} \mathbf{x} \right]_{+} \right\|_{\max}$$

where $[\]_{+}$ denotes the positive parts of the components of a real vector. Then

$$G(\mathbf{x}) := \mathbb{E} \left(\left\| \left[\mathbf{d} - T \mathbf{x} \right]_{+} \right\|_{\max} \right)$$

represents the *expected shortfall*, and the constraint $G(\mathbf{x}) \leq \Gamma$ is a *joint-form integrated chance constraint* as defined by Klein Haneveld (1986). (We must mention that in finance, the term *expected shortfall* is used in meanings different from the above. Financial definitions involve an α -quantile and conditional expectation.)

Remark 10 Since we have set $\mathbf{q} = \mathbf{0}$, it follows that Assumption 6 obviously holds for any $w > 0$, i.e., $f_w(\mathbf{x}, \boldsymbol{\xi}_{\omega}) = wg(\mathbf{x}, \boldsymbol{\xi}_{\omega})$ holds for $\mathbf{x} \in \mathbb{R}^n$, $\omega \in \Omega$. It follows that we have $F_w(\mathbf{x}) = wG(\mathbf{x})$ ($\mathbf{x} \in \mathbb{R}^n$).

Klein Haneveld examines two problems. The first is an integrated chance constrained problem that he denotes by $ICC_3(\Gamma)$ and defines for $\Gamma \geq 0$:

$$\begin{aligned} & \min \mathbf{c}^T \mathbf{x} \\ & \text{subject to} \\ & \mathbf{x} \in X, \\ & \mathbb{E} \left(\left\| \left[\mathbf{d} - T \mathbf{x} \right]_{+} \right\|_{\max} \right) \leq \Gamma. \end{aligned}$$

The second is a recourse problem that Klein Haneveld denotes by $L_3(w)$ and defines for $w \geq 0$:

$$\begin{aligned} & \min \mathbf{c}^T \mathbf{x} + w \mathbb{E} \left(\left\| \left[\mathbf{d} - T \mathbf{x} \right]_{+} \right\|_{\max} \right) \\ & \text{subject to} \\ & \mathbf{x} \in X. \end{aligned}$$

Klein Haneveld proves that the problems $ICC_3(\Gamma)$ and $L_3(w)$ are in a sense equivalent for certain (Γ, w) pairs.

To establish the connection with the relaxed model, let us restrict our attention to the case $w > 0$. Our relaxed model incorporates both the integrated chance constraint $G(\mathbf{x}) \leq \Gamma$ occurring in $ICC_3(\Gamma)$; and the penalty term $wG(\mathbf{x})$ occurring in $L_3(w)$. (We have $wG(\mathbf{x}) = F_w(\mathbf{x})$ according to Remark 10.) The equivalence established by Klein Haneveld implies that either the integrated chance constraint or the penalty term may be redundant in this special form of the relaxed model. Both may be needed, however, from a practical point of view.

Let us now return to the system design interpretation of the stochastic model. The first-stage variables represent system design; and the second-stage variables

describe the operation of the system under certain circumstances. Here the auxiliary vector \mathbf{s} represents external resources consumed by the system. The weight w represents the unit price of these external resources. The interpretation of the relaxed model depends on whether Assumption 6 holds.

Economic interpretation, expensive resources. Suppose that w is large and hence Assumption 6 holds. Then the constraint $G(\mathbf{x}) \leq \Gamma$ expresses a restriction on the expected amount of external resources consumed. Such constraints can be used to avoid over-dependence on certain resources. For example, in case of modeling energy supplies of an economy, political aspects can be taken into account.

Economic interpretation, cheap resources. Suppose that w is small and hence Assumption 6 does not hold. (Even Assumption 1 may not hold.) Then the system takes advantage of the low price of the external resources by consuming more of them than inevitably required. This is the normal mode of operation, described by the optimal solution of problem (4).

Suppose, however, that temporary shortages in external resources are expected to occur from time to time. The system must remain operational in case a temporary restriction on external resources should be imposed sometime in the future. During shortages, the system switches to *emergency* operation mode, described by an optimal solution of problem (6). The constraint $G(\mathbf{x}) \leq \Gamma$ expresses a restriction on the expected amount of external resources consumed in emergency operation mode. The parameter Γ may depend on the estimated lengths of shortages and on storage capacity, i.e., the magnitude of the reserve of external resources that can be aggregated during periods of normal supply.

Operation costs of the normal mode are expressed by the term $F_w(\mathbf{x})$. Emergency operation mode is more expensive. (During a shortage, the system can not take advantage of the low price of external resources.) Suppose, however, that normal operation is possible in most of the time. Then the overall operation cost remains $F_w(\mathbf{x})$.

In this interpretation, (8) is not a ‘traditional’ two-stage recourse model.

Remark 11 In these latter economic models, both the parameter Γ and the weight w are determined by factors outside the scope of the models. The weight may be determined e.g., by market equilibrium.

Given a certain weight w , the decision maker needs to know whether the first or the second interpretation is relevant. Let \mathbf{x}^* be an (approximate) optimal solution offered by some solution process. If Assumption 6 holds in \mathbf{x}^* , then the first interpretation is relevant.

Checking Assumption 6 in a certain point causes no theoretical difficulty if the distribution is finite discrete.

2 The Level methods

The Level Method and the Constrained Level Method were proposed by Lemaréchal et al. (1995). These are bundle-type methods that can be presented in simple form, and proved very effective in practice. Fábíán (2000) proposed

inexact versions of these methods. First we sketch the exact methods, then describe the inexact versions. In order to avoid duplication, details will be given in the second part.

The Level Method solves the problem

$$\begin{aligned} & \min \phi(\mathbf{x}) \\ & \text{subject to} \\ & \mathbf{x} \in X, \end{aligned} \tag{9}$$

where $X \subset \mathbb{R}^n$ is a convex bounded polyhedron, and ϕ a real-valued convex function, Lipschitzian relative to X . This is an iterative method, a direct generalization of the classical cutting-plane method. A *cutting-plane model* of ϕ is maintained using function values and subgradients computed at the known iterates. This is the upper cover of the linear support functions drawn at the known iterates. Hence this is a polyhedral convex lower approximation of ϕ .

The level sets of this model function are rather stable, and can be used for regularization. The next iterate is obtained by *projecting* the current iterate onto a certain level set of the current model function. (By projecting, we mean finding the point of the level set closest to the current iterate.) Each iteration requires the solution of a linear programming and a convex quadratic programming problem. The linear problem determines the level set to be used for regularization. The quadratic problem determines the projection of the current iterate onto this level set.

A lower bound for the optimum of the problem (9) is obtained by minimizing the model function over X . An upper bound is, on the other hand, the best function value found. The difference between the upper and lower bounds is called the *gap*. Lemaréchal et al. prove the following efficiency estimate: To obtain a gap smaller than ϵ , it suffices to perform

$$\kappa \left(\frac{D\Lambda}{\epsilon} \right)^2 \tag{10}$$

iterations, where D is the diameter of the feasible polyhedron, Λ is a Lipschitz constant of ϕ , and κ is a constant that depends only on the parameter of the algorithm.

The Constrained Level Method solves the problem

$$\begin{aligned} & \min \phi(\mathbf{x}) \\ & \text{subject to} \\ & \mathbf{x} \in X, \\ & \psi(\mathbf{x}) \leq 0, \end{aligned} \tag{11}$$

where $X \subset \mathbb{R}^n$ is a convex bounded polyhedron, and ϕ, ψ are real-valued convex functions, Lipschitzian relative to X . For the sake of simplicity of the discussion we also assume that ψ is non-negative. (This can always be supposed

for non-differentiable problems.) Hence the last constraint in (11) actually means $\psi(\mathbf{x}) = 0$.

Let O^* denote the optimal objective value of the above problem. If O^* is known in advance, then the quality of an approximate solution $\mathbf{x} \in X$ can be measured by

$$e(\mathbf{x}) = \max \{ \phi(\mathbf{x}) - O^*, \psi(\mathbf{x}) \}.$$

For example, $e(\mathbf{x}) = 0$ means that \mathbf{x} is optimal. The original problem is equivalent to the convex problem

$$\begin{aligned} & \min e(\mathbf{x}) \\ & \text{subject to} \\ & \mathbf{x} \in X. \end{aligned} \tag{12}$$

The Lagrangian dual of the above problem can be written as

$$\max_{0 \leq \alpha \leq 1} h(\alpha), \tag{13}$$

where

$$h(\alpha) = \min_{\mathbf{x} \in X} \alpha (\phi(\mathbf{x}) - O^*) + (1 - \alpha)\psi(\mathbf{x}).$$

The Constrained Level Method is an iterative method that maintains cutting-plane models of ϕ and ψ . Using these model functions, a lower approximation is computed for O^* . Moreover, a cutting-plane model is constructed for the dual objective function. This is a polyhedral concave upper approximation of h . The maximum of this upper approximating function over the $[0, 1]$ interval is obviously an upper bound for the maximum of the dual problem (13). The latter maximum is 0 from duality. Hence the aim is to direct the search for new iterates in such a manner that the maximum of the upper approximating function decreases beyond some preset stopping tolerance. If this is achieved, then a near-optimal solution can be constructed.

The Constrained Level Method is a primal-dual method. Given a dual iterate $\alpha \in [0, 1]$, the primal iterate $\mathbf{x} \in X$ is selected by applying an unconstrained-Level-Method-type iteration to the convex function $\alpha\phi + (1 - \alpha)\psi$. The dual iterate is left unchanged as long as possible. Hence the Constrained Level Method consists of runs of the Level Method.

The method requires no Slater assumption and the efficiency is not affected by large Lagrange multipliers. Lemaréchal, Nemirovskii, and Nesterov prove the following efficiency estimate: To obtain an ϵ -optimal solution (i.e. $\mathbf{x}^* \in X$ satisfying $e(\mathbf{x}^*) \leq \epsilon$), it suffices to perform

$$\kappa \left(\frac{D\Delta}{\epsilon} \right)^2 \ln \left(\frac{D\Delta}{\epsilon} \right) \tag{14}$$

iterations, where D is the diameter of the feasible polyhedron, Λ is a common Lipschitz constant of ϕ and ψ , and κ is a constant that depends only on the parameters of the method.

Lemaréchal et al. report on successful application of the above methods to a variety of problems. The experimental results suggest much better practical behavior than the above estimates. Another factor of efficiency is the computational effort of a single iteration. As mentioned earlier, each iteration requires the solution of a linear programming and a convex quadratic programming problem. The current sizes of these problems depend on the size of the *bundle*, i.e., on the amount of the information used to construct the current approximating models. For the unconstrained case, Lemaréchal et al. propose a bundle reduction strategy that preserves the theoretical efficiency estimate of the Level Method. (The cutting planes that are not active in the evaluation of the current gap can be omitted after certain iterations.) Lemaréchal et al. also report on the practical behavior of the reduction strategy. In their experiments, bundle reduction caused no substantial increase in the number of iterations. Moreover, the size of the bundle never exceeded $2n$, where n is the dimension of the space in which the optimization is performed. Although tests were made with the Level Method, the results are instructive for the constrained case, because the constrained method consists of runs of the unconstrained method.

The inexact methods only differ from the exact methods in the use of accuracy tolerances that we denote by δ . Let us consider the convex programming problem (9), and assume that given a point $\check{\mathbf{x}} \in X$ and some accuracy of $\delta > 0$, we can construct a linear function l satisfying

$$l \leq \phi \quad \text{and} \quad \phi(\check{\mathbf{x}}) - l(\check{\mathbf{x}}) \leq \delta.$$

The gradient of this linear function is clearly a δ -subgradient of ϕ at $\check{\mathbf{x}}$. (Basic theory of approximate subgradients can be found in Lemaréchal (1982).) We will say that l is a δ -support function of ϕ at $\check{\mathbf{x}}$. Assume, moreover, that for any possible $\check{\mathbf{x}}$ and δ , our δ -support functions satisfy the Lipschitz condition with a constant not depending on $\check{\mathbf{x}}$ or δ .

The Inexact Level Method works in the following way. Suppose that given the point \mathbf{x}_ℓ and accuracy δ_ℓ , we constructed a δ_ℓ -support function l_ℓ ($\ell = 1, \dots, k$). The cutting-plane model of ϕ is

$$\phi_k(\mathbf{x}) = \max_{1 \leq \ell \leq k} l_\ell(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^n).$$

A lower and an upper bound for the optimum is computed as

$$v_k = \min_{\mathbf{x} \in X} \phi_k(\mathbf{x}) \quad \text{and} \quad \tau_k = \min_{1 \leq \ell \leq k} \phi_k(\mathbf{x}_\ell) + \delta_\ell. \quad (15)$$

The gap between the above bounds is $\Delta_k = \tau_k - v_k$.

Let $0 < \lambda < 1$ be some preset parameter. The next iterate \mathbf{x}_{k+1} is computed by projecting the current iterate \mathbf{x}_k onto the level set $\{\mathbf{x} \mid \phi_k(\mathbf{x}) \leq \nu_k + \lambda \Delta_k\}$, i.e., by the solution of the convex quadratic problem

$$\begin{aligned} & \min \|\mathbf{x} - \mathbf{x}_k\|^2 \\ & \text{subject to} \\ & \mathbf{x} \in X, \\ & \phi_k(\mathbf{x}) \leq \nu_k + \lambda \Delta_k. \end{aligned} \tag{16}$$

The accuracy parameter for the next cutting plane is prescribed as $\delta_{k+1} = \gamma \Delta_k$, where γ is a constant parameter satisfying $0 < \gamma < (1 - \lambda)^2$.

Fábíán (2000) proved that the efficiency estimate (10) also holds for the Inexact Level Method. (Only the constant κ is different from that of the exact method, and depends on the parameters λ and γ .) The proof follows the steps of the proof for the exact method: starting from \mathbf{x}_1 , let us consider the maximal sequence of iterations $\mathbf{x}_1 \rightarrow \mathbf{x}_2, \dots, \mathbf{x}_{s-1} \rightarrow \mathbf{x}_s$, at the end of which the gap ‘has not been reduced much’, i.e., the following inequalities hold:

$$(1 - \lambda)\Delta_1 \leq \Delta_s, \quad (1 - \lambda)\Delta_1 > \Delta_{s+1}.$$

The iteration $\mathbf{x}_s \rightarrow \mathbf{x}_{s+1}$ will be called *critical*. The above construction is repeated starting from \mathbf{x}_{s+1} . Thus the iterations are grouped into finite sequences, and the sequences are separated with critical iterations. The number of the sequences cannot be large: Consider the gap at the start of each sequence; these gaps are decreasing at an exponential rate (the quotient being $1 - \lambda$). Moreover, it turns out that each sequence has its respective stability center; a feasible point towards which the iterates of this sequence are attracted. This crucial property allows the computation of upper bounds on the lengths of the sequences.

Remark 12 The method works as well with a more general selection of ν_k instead of the minimum defined in (15). The efficiency estimate remains valid if ν_k satisfies the following two requirements: $\nu_k \leq \tau_k$, and $\phi_k(\check{\mathbf{x}}) \leq \nu_k$ for some $\check{\mathbf{x}} \in X$.

Let us now consider the constrained convex programming problem (11). The Inexact Constrained Level Method works in the following way. Suppose that given the point \mathbf{x}_ℓ and the accuracy δ_ℓ , we constructed the δ_ℓ -support function l_ℓ for ϕ ; and l_ℓ^\sharp for ψ ($\ell = 1, \dots, k$). The cutting-plane models of ϕ and ψ will be

$$\phi_k(\mathbf{x}) = \max_{1 \leq \ell \leq k} l_\ell(\mathbf{x}) \quad \text{and} \quad \psi_k(\mathbf{x}) = \left[\max_{1 \leq \ell \leq k} l_\ell^\sharp(\mathbf{x}) \right]_+ \quad (\mathbf{x} \in \mathbb{R}^n),$$

where $[\cdot]_+$ means the positive part of a real number. A lower approximation for the optimum O^* can be computed as

$$O_k = \min \{ \phi_k(\mathbf{x}) \mid \mathbf{x} \in X, \psi_k(\mathbf{x}) \leq 0 \}.$$

Moreover, a polyhedral concave upper approximating function of the dual objective function h can be constructed as

$$h_k(\alpha) = \min_{1 \leq \ell \leq k} \alpha (\phi_k(\mathbf{x}_\ell) + \delta_\ell - O_k) + (1 - \alpha) (\psi_k(\mathbf{x}_\ell) + \delta_\ell) \quad (0 \leq \alpha \leq 1).$$

Its maximum,

$$H_k = \max_{0 \leq \alpha \leq 1} h_k(\alpha) \tag{17}$$

is an upper bound for the maximum of the dual problem (13) that is 0. The aim is to direct the search for new iterates in such a manner that H_k decreases beyond some preset stopping tolerance $\epsilon > 0$. If this is achieved, then a near-optimal solution can be constructed in the form of a convex combination of former iterates:

$$\mathbf{x}^* = \sum_{\ell=1}^k \varrho_\ell \mathbf{x}_\ell. \tag{18}$$

A viable convex combination can be found through the solution of the following linear programming problem, the variables of which are $\varrho_1, \dots, \varrho_k$, and ϑ :

$$\begin{aligned} & \min \vartheta \\ & \text{subject to} \\ & \sum_{\ell=1}^k \varrho_\ell (\phi_k(\mathbf{x}_\ell) + \delta_\ell) - O_k \leq \vartheta, \\ & \sum_{\ell=1}^k \varrho_\ell (\psi_k(\mathbf{x}_\ell) + \delta_\ell) \leq \vartheta, \\ & \sum_{\ell=1}^k \varrho_\ell = 1, \\ & \varrho_1, \dots, \varrho_k \geq 0. \end{aligned}$$

The dual of the above problem is equivalent to the one-dimensional maximization problem (17). Hence in the optimal solution we have $\vartheta = H_k$. From the convexity of the feasible polyhedron X , and of the functions ϕ and ψ , it follows that the convex combination (18) is near-optimal with the tolerance H_k , i.e. $e(\mathbf{x}^*) \leq H_k$.

The iterates are selected in the following way: first a dual iterate α_k is selected such that $h_k(\alpha_k)$ is ‘sufficiently close’ to H_k . Consider the interval $I_k = [\underline{\alpha}_k, \bar{\alpha}_k] \subseteq [0, 1]$ on which h_k is non-negative. Let the subinterval $\hat{I}_k \subset I_k$ be obtained by shrinking I_k towards its center with the factor $(1 - \mu)$, where $0 < \mu < 1$ is some preset parameter. Owing to the concavity of h_k , it easily follows that

$$h_k(\alpha) \geq \frac{\mu}{2} H_k \tag{19}$$

holds for any $\alpha \in \hat{I}_k$. Specifically, the dual iterate is selected as follows: let α_1 be the center of the interval I_1 . For $k > 1$, the dual iterate is left unchanged if possible, i.e., if the former iterate α_{k-1} falls into \hat{I}_k , then let $\alpha_k = \alpha_{k-1}$. Otherwise let α_k be the center of the interval I_k .

After this, the primal iterate \mathbf{x}_{k+1} is selected by applying an Inexact-Level-Method-type iteration to the convex function $\alpha_k\phi + (1-\alpha_k)\psi$. For $1 \leq \ell \leq k$, the linear function $\alpha_k l_\ell + (1-\alpha_k)l_\ell^\#$ is a δ_ℓ -support function at \mathbf{x}_ℓ . Hence $\alpha_k\phi_k + (1-\alpha_k)\psi_k$ is an appropriate cutting-plane model of $\alpha_k\phi + (1-\alpha_k)\psi$. According to the construction of the Inexact Level Method, we get $\tau_k = \min_{1 \leq \ell \leq k} \alpha_k\phi_k(\mathbf{x}_\ell) + (1-\alpha_k)\psi_k(\mathbf{x}_\ell) + \delta_\ell$. The lower level is selected specially as $\nu_k = \alpha_k O_k$. (The selection is in accordance with Remark 12.) This gives the gap $\Delta_k = \tau_k - \nu_k = h_k(\alpha_k)$. Hence the next primal iterate \mathbf{x}_{k+1} will be the projection of the former iterate \mathbf{x}_k onto the level set

$$\{\mathbf{x} \in X \mid \alpha_k\phi_k(\mathbf{x}) + (1-\alpha_k)\psi_k(\mathbf{x}) \leq \alpha_k O_k + \lambda h_k(\alpha_k)\}.$$

The accuracy parameter for the next cutting plane is prescribed as $\delta_{k+1} = \gamma \Delta_k$, where γ is a constant parameter satisfying $0 < \gamma < (1-\lambda)^2$.

This procedure is iterated until the gap $h_k(\alpha_k)$ becomes small enough. (Inequality (19) ensures that H_k decreases with the gap.) Since the dual iterate is left unchanged as long as possible, the method consists of runs of the Inexact Level Method.

Fábián (2000) proved that the efficiency estimate (14) also holds for the inexact method. (Only the constant κ is different from that of the exact method, and depends on the parameters λ, μ and γ .)

By setting the tolerances δ to 0 in the above discussion, the reader gets a description of the exact methods of Lemaréchal et al. The inexact methods inherit the stability of the exact methods.

The bundle reduction technique proposed by Lemaréchal et al. can be adapted to the constrained method. After critical iterations, we can omit the cutting planes that are inactive in the computation of each of the following objects: the current approximation O_k of the optimum, the current dual iterate α_k , and the current upper level τ_k . (After reduction, the size of the bundle is at most $n + 12$.) This reduction strategy preserves the theoretical efficiency estimates, but it is heuristic in the sense that there is no theoretical bound on the size of the bundle between critical iterations.

The quadratic problems to be solved in course of the method are special ones. Namely, they are *least squares problems* for which several numerically efficient algorithms exist, see e.g. Björck (1996).

3 Level decomposition

We apply the Inexact Constrained Level Method to the standard model (7) and to the relaxed model (8). The relevant convex programming problems can be obtained by substituting

$$\phi(\mathbf{x}) := \mathbf{c}^T \mathbf{x} + F(\mathbf{x}), \quad \psi(\mathbf{x}) := G(\mathbf{x}), \quad (20)$$

and

$$\phi(\mathbf{x}) := \mathbf{c}^T \mathbf{x} + F_w(\mathbf{x}), \quad \psi(\mathbf{x}) := [G(\mathbf{x}) - \Gamma]_+, \quad (21)$$

respectively, in the general constrained convex programming problem (11). Function information of appropriate accuracy will be provided by an *oracle* to be described in this section. In the case of the relaxed model, the parameters w and Γ are assumed to be determined. Inexact function data means that given $\check{\mathbf{x}}$, we approximately compute the exact quantities $F_w(\check{\mathbf{x}})$ and $G(\check{\mathbf{x}}) - \Gamma$.

The framework of the method used for the solution of the master problem is the following:

Initialize.

Set the stopping tolerance $\epsilon > 0$.

Set the parameters λ, μ , and γ ($0 < \lambda, \mu < 1$; $0 \leq \gamma < (1 - \lambda)^2$).

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

Set the starting accuracy $\delta_1 > 0$.

Let $k := 1$ (iteration counter).

Update bundle.

Given the point \mathbf{x}_k and the accuracy δ_k ,

call the oracle to construct the δ_k -support functions l_k and $\tilde{l}_k^\#$ for ϕ and ψ , respectively.

Define the upper covers $\phi_k(\mathbf{x}) := \max_{1 \leq \ell \leq k} l_\ell(\mathbf{x})$, $\psi_k(\mathbf{x}) := \left[\max_{1 \leq \ell \leq k} \tilde{l}_\ell^\#(\mathbf{x}) \right]_+$.

Compute the lower approximation of the optimum

$O_k := \min \{ \phi_k(\mathbf{x}) \mid \mathbf{x} \in X, \psi_k(\mathbf{x}) \leq 0 \}$.

Define the function $h_k(\alpha) := \min_{1 \leq \ell \leq k} \alpha(\phi_k(\mathbf{x}_\ell) - O_k) + (1 - \alpha)\psi_k(\mathbf{x}_\ell) + \delta_\ell$,

and compute its maximum $H_k := \max_{0 \leq \alpha \leq 1} h_k(\alpha)$.

Check for optimality.

If $H_k < \epsilon$, then near-optimal solution found;

compute best point according to (18), and stop.

Find dual iterate.

Determine the interval $I_k = [\underline{\alpha}_k, \bar{\alpha}_k] \subseteq [0, 1]$ on which h_k takes non-negative values. Let $|I_k|$ denote the length of I_k .

Compute α_k :

- for $i = 1$, let $\alpha_1 := \frac{1}{2}(\bar{\alpha}_1 + \underline{\alpha}_1)$,
- for $i > 1$, let $\alpha_k := \begin{cases} \alpha_{k-1}, & \text{if } \underline{\alpha}_k + \frac{\mu}{2}|I_k| \leq \alpha_{k-1} \leq \bar{\alpha}_k - \frac{\mu}{2}|I_k|, \\ \frac{1}{2}(\bar{\alpha}_k + \underline{\alpha}_k), & \text{otherwise.} \end{cases}$

Find primal iterate.

Let \mathbf{x}_{k+1} be the optimal solution of the convex quadratic programming problem

$$\begin{aligned} & \min \|\mathbf{x} - \mathbf{x}_k\|^2 \\ & \text{subject to} \\ & \quad \mathbf{x} \in X, \\ & \quad \alpha_k \phi_k(\mathbf{x}) + (1 - \alpha_k) \psi_k(\mathbf{x}) \leq \alpha_k O_k + \lambda h_k(\alpha_k). \end{aligned}$$

Loop.

Let $\delta_{k+1} := \gamma h_k(\alpha_k)$.

Increment k .

→ *Update bundle.*

This method regularizes both feasibility and optimality issues. Moreover the size of the master problems can be controlled by the bundle reduction technique described at the end of previous section¹. The γ parameter controls the accuracy prescribed for the oracle. (If we set $\gamma = 0$ in Initialization, then we get the exact method.)

In the remaining part of this section, we deal with the operation of the oracle. With the notations of the standard model, we describe the construction of respective δ -support functions for the functions $F(\mathbf{x})$ and $G(\mathbf{x})$ at a given point $\check{\mathbf{x}}$. In the case of the relaxed model the procedure is the same, only the notation differs: (21) needs to be used instead of (20).

Effort can be spared by two means: by approximation of the distribution, and by approximate solution of the recourse problems. Both means are optional. In fact the distribution can only be approximated under the dimensionality assumption (22) to be formulated in the forthcoming subsection; however, discretization is needed if the original distribution is continuous. If the representation (22) does not hold, then a Sample Average Approximation (SAA) scheme can be applied. Such schemes are mentioned in section 7.

We employ the two kinds of approximation in a proportion controlled by the constant parameter $0 \leq \theta \leq 1$: with $\theta = 1$, the original distribution is used throughout; with $\theta = 0$, the recourse problems are solved exactly. (The recourse problems are solved using a linear programming method that returns a primal feasible and a dual feasible solution with a duality gap less than $\tilde{\delta} = \theta \delta$.)

3.1 Approximation of the distribution

In order to alleviate notation, we assume in this subsection that $T_\omega = T$ ($\omega \in \Omega$) holds. Since only the right-hand-side vector \mathbf{d} is random, we have $\boldsymbol{\xi} \equiv \mathbf{d}$. We will use the notation $\boldsymbol{\xi}$. However, the forthcoming discussion can be easily extended to allow a random T matrix.

¹ Although we implemented this bundle reduction technique, it was never used in course of the solution of our stochastic test problems. In our test runs, the number of the master iterations was always much less than the number of the variables in the master problem. (Our test results are reported in section 6.)

We assume moreover that our random parameters can be traced back to but a few random parameters. Namely, we assume that ξ linearly depends on a low-dimensional random vector, i.e.,

$$\xi_\omega = S\eta_\omega \quad (\omega \in \Omega), \quad (22)$$

where S is an $r \times \tilde{r}$ -matrix, and η_ω ($\omega \in \Omega$) are realizations of an \tilde{r} -dimensional random vector η .

Nearness in the support of η can be exploited by applying the classic discretization and bounding methods of Kall (1980), Kall and Stoyan (1982), Birge and Wets (1986), Frauendorfer and Kall (1988), and Frauendorfer (1988). Although these methods were originally devised for the discrete approximation of a continuous distribution, they can as well be applied to approximate a discrete distribution with a coarser one. Mayer (1998) have successfully applied discretization methods for this purpose.

Discretization and bounding methods partition the support of η into cells. The weight of a cell C is defined as

$$p_C := \mathbb{P}(\eta \in C).$$

We assume there are no empty cells. The conditional expectation

$$F_C(\mathbf{x}) := \mathbb{E}(f(\mathbf{x}, S\eta) \mid \eta \in C) \quad (23)$$

is considered as a function of $\mathbf{x} \in \mathbb{R}^n$. Obviously

$$F(\mathbf{x}) = \sum_C p_C F_C(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^n).$$

Based on the cell structure, two coarsely distributed discrete random vectors are constructed instead of η ; namely $\underline{\eta}$ for lower approximation, and $\bar{\eta}$ for upper approximation. The construction is such that $\underline{\eta}$ yields lower approximating functions

$$\underline{F}_C(\mathbf{x}) \leq F_C(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^n)$$

for each cell C ; and $\bar{\eta}$ yields upper approximating functions

$$\bar{F}_C(\mathbf{x}) \geq F_C(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^n).$$

Lower and upper approximating functions for the expected recourse function are constructed as

$$\underline{F}(\mathbf{x}) = \sum_C p_C \underline{F}_C(\mathbf{x}) \quad \text{and} \quad \bar{F}(\mathbf{x}) = \sum_C p_C \bar{F}_C(\mathbf{x}).$$

Traditionally, approximation schemes are used in the following framework. An approximating problem is constructed instead of the problem (7) by replacing the expected recourse function with a lower approximating function $\underline{F}(\mathbf{x})$. The approximating problem is solved, let \mathbf{x}^* denote the optimal solution. The upper approximating function $\overline{F}(\mathbf{x})$ is evaluated at \mathbf{x}^* . If the difference $\overline{F}(\mathbf{x}^*) - \underline{F}(\mathbf{x}^*)$ is not significant, then the procedure stops. Otherwise the approximating coarse distributions are refined, and the above procedure is iterated. Hence the traditional framework prescribes the (exact) solution of a series of approximate problems. The objective functions of these problems are successive approximations of the true objective function.

In the traditional framework, effort may be wasted on optimization (and hence on expensive function evaluations) when the approximation is not accurate. In a typical implementation of the traditional framework, the main module is the distribution approximation scheme, and the submodule is the two-stage stochastic problem solver. To what extent the approximation should be refined between solver calls, is a difficult question. The following observation shows the intricacy and importance of this question.

Let us observe that even the traditional framework makes use of inexact cuts. The solution of each approximate problem may start with a cutting-plane model of the expected recourse function inherited from the predecessor problem. Since the predecessor problem has been constructed using an approximation of the distribution coarser than the current one, the inherited cutting-plane model contains cuts that are inexact in the current problem. Alternatively, one might either build a cutting-plane model from scratch each time one sets to solve an approximate problem, or recompute all existing cuts after each refinement of the distribution approximation. Both of these alternatives demand considerable effort.

To control accuracy in the traditional framework, Mayer (1998) proposed a heuristic method.

In contrast to the traditional framework, the Level Decomposition Method solves the single problem (7). The approximating coarse distributions are refined in accordance with the solution process of the master problem.

In course of the present project we adapted the simplest bounding methods to the Level Decomposition framework. Our cells are \tilde{r} -dimensional intervals.

The lower approximating distribution is constructed using Jensen’s inequality: $\underline{\eta}$ has as many realizations as the number of the cells. The realization corresponding to cell C is constructed by concentrating into the barycenter the weight of those η realizations that fall into the cell C . In order to ease the forthcoming discussion, let us introduce the notation $P(\mathbf{x}, \eta)$ and $P^\sharp(\mathbf{x}, \eta)$ for the following special forms of the problems (4) and (6), respectively:

$$\begin{array}{ll}
 \min \mathbf{q}^T \mathbf{y} + w \| \mathbf{s} \|_1 & \min \| \mathbf{s} \|_1 \\
 \text{subject to} & \text{subject to} \\
 T \mathbf{x} + W \mathbf{y} + \mathbf{s} = S \eta, & T \mathbf{x} + W \mathbf{y} + \mathbf{s} = S \eta, \\
 \mathbf{y} \geq \mathbf{0}, & \mathbf{y} \geq \mathbf{0}.
 \end{array}$$

For each cell C and any \mathbf{x} , the lower approximating function value $\underline{F}_C(\mathbf{x})$ is defined as the optimum of the problem $P(\mathbf{x}, \underline{\eta}_C)$, where $\underline{\eta}_C := \mathbb{E}(\eta \mid \eta \in C)$ is the realization of $\bar{\eta}$ corresponding to C . In order to get an estimate of $\underline{F}_C(\check{\mathbf{x}})$, we solve the problem $P(\check{\mathbf{x}}, \underline{\eta}_C)$ approximately, setting the tolerance of the LP duality gap to $\tilde{\delta} = \theta\delta$. Let \mathbf{z}_C denote the (near-optimal) dual feasible solution returned by the linear programming method. Let us define the linear functions

$$l_C(\mathbf{x}) = \mathbf{z}_C^T (S\underline{\eta}_C - T\mathbf{x}) \quad \text{and} \quad l(\mathbf{x}) = \sum_C p_C l_C(\mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^n).$$

We have $l_C(\mathbf{x}) \leq \underline{F}_C(\mathbf{x})$ since \mathbf{z}_C is a dual feasible solution of the problem $P(\mathbf{x}, \underline{\eta}_C)$ for any \mathbf{x} . It follows that $l(\mathbf{x}) \leq \underline{F}(\mathbf{x})$ for any \mathbf{x} . Moreover we have $\|\mathbf{z}_C\|_{\max} \leq w$ owing to the dual feasibility of \mathbf{z}_C . Hence in Euclidean norm, $\|\mathbf{z}_C\| \leq \sqrt{r}w$. It follows that $l_C(\mathbf{x})$ and hence $l(\mathbf{x})$ are Lipschitz continuous with the constant $\sqrt{r}w\|T\|$. With the values $L_C = l_C(\check{\mathbf{x}})$ and $L = l(\check{\mathbf{x}})$ obviously we have

$$L_C \leq \underline{F}_C(\check{\mathbf{x}}) \leq L_C + \tilde{\delta} \quad \text{and hence} \quad L \leq \underline{F}(\check{\mathbf{x}}) \leq L + \tilde{\delta}. \tag{24}$$

The upper approximating distribution is constructed using the Edmundson-Madansky inequality, hence the realizations of $\bar{\eta}$ correspond to the vertices of the cells. Let $\bar{\eta}_C^j$ ($j = 1, \dots, 2^{\bar{r}}$) denote the realizations of $\bar{\eta}$ corresponding to the vertices of C . By definition, $\bar{F}_C(\mathbf{x})$ is a certain convex combination of the optimal objective values of the problems $P(\mathbf{x}, \bar{\eta}_C^j)$. In order to get an estimate of $\bar{F}_C(\check{\mathbf{x}})$, we approximately solve the problems $P(\check{\mathbf{x}}, \bar{\eta}_C^j)$ with a duality gap less than $\tilde{\delta}$. Let $(\mathbf{y}_C^j, \mathbf{s}_C^j)$ and \mathbf{z}_C^j denote the (near-optimal) primal and dual feasible solutions, respectively, that the linear programming method returns. Let

$$U_C = \sum_{j=1}^{2^{\bar{r}}} \varrho_C^j \left(\mathbf{q}^T \mathbf{y}_C^j + w \|\mathbf{s}_C^j\|_1 \right) \quad \text{and} \quad U = \sum_C p_C U_C,$$

where the coefficients ϱ_C^j are the same as those of the convex combination that produced $\bar{F}_C(\check{\mathbf{x}})$. Obviously we have

$$U_C \geq \bar{F}_C(\check{\mathbf{x}}) \geq U_C - \tilde{\delta} \quad \text{and hence} \quad U \geq \bar{F}(\check{\mathbf{x}}) \geq U - \tilde{\delta}.$$

Comparing this to (24) we get

$$U - L \geq \bar{F}(\check{\mathbf{x}}) - \underline{F}(\check{\mathbf{x}}) \geq U - L - 2\tilde{\delta}. \tag{25}$$

If $\delta \geq U - L$ holds, then $l(\mathbf{x})$ is a δ -support function of the expected recourse function $F(\mathbf{x})$ at $\check{\mathbf{x}}$. Moreover $l(\mathbf{x})$ is Lipschitz-continuous with a constant not depending on $\check{\mathbf{x}}$ or δ .

In case of the function $G(\mathbf{x})$, we construct the linear functions $l_C^\sharp(\mathbf{x}), l^\sharp(\mathbf{x})$, and the bounds $L_C^\sharp, L^\sharp, U_C^\sharp, U^\sharp$ in a similar way, through approximate solution of problems of the type $P^\sharp(\mathbf{x}, \boldsymbol{\eta})$. If both $\delta \geq U - L$ and $\delta \geq U^\sharp - L^\sharp$ holds, then the oracle returns l as a δ -support function of F , and l^\sharp as a δ -support function of G . (The cell structure is retained for use in further oracle calls.) Otherwise the approximation is refined. We subdivide the cell that maximizes the product

$$p_C \max \left\{ U_C - L_C - \delta, U_C^\sharp - L_C^\sharp - \delta \right\}.$$

The above product should be positive for some cells, otherwise the oracle would have returned the current support functions. Let \check{C} denote the cell selected for subdivision, and $\bar{\boldsymbol{\eta}}_C^j$ ($j = 1, \dots, 2^{\check{r}}$) the realizations of $\bar{\boldsymbol{\eta}}$ corresponding to the vertices of \check{C} .

The coordinate along which to cut is chosen as follows: Let \mathbf{e}_t ($t = 1, \dots, \check{r}$) denote the unit vectors in the space of the $\boldsymbol{\eta}$ values. Having t fixed, the edges of \check{C} parallel to \mathbf{e}_t are marked, i.e. the edge between the j th and j' th vertices is marked if $\bar{\boldsymbol{\eta}}_C^j - \bar{\boldsymbol{\eta}}_C^{j'} = u\mathbf{e}_t$ holds. The optimal objective value of the recourse problem $P(\check{\mathbf{x}}, \boldsymbol{\eta})$ is considered as a function of $\boldsymbol{\eta}$, and nonlinearity of this function is measured along the marked edges. A possible measure of nonlinearity along the (j, j') edge is based on function values and \mathbf{e}_t -directional derivatives at the endpoints. Since we have solved the relevant recourse problems approximately, we only have approximate function values, namely

$$v = \left(S\bar{\boldsymbol{\eta}}_C^j - T\check{\mathbf{x}} \right)^T \mathbf{z}_C^j \quad \text{and} \quad v' = \left(S\bar{\boldsymbol{\eta}}_C^{j'} - T\check{\mathbf{x}} \right)^T \mathbf{z}_C^{j'}.$$

Approximations of the directional derivatives are computed as

$$\varsigma = \mathbf{e}_t^T S^T \mathbf{z}_C^j \quad \text{and} \quad \varsigma' = \mathbf{e}_t^T S^T \mathbf{z}_C^{j'}.$$

To simplify notation, let us transform the interval $[\bar{\boldsymbol{\eta}}_C^j, \bar{\boldsymbol{\eta}}_C^{j'}]$ into $[0, u]$, and define the $[0, u] \rightarrow \mathbb{R}$ function $\chi(o) = \max \{ v + \varsigma o, v' + \varsigma'(o - u) \}$. Moreover, let π denote the $[0, u] \rightarrow \mathbb{R}$ linear function whose graph contains the points $(0, v)$ and (u, v') . We define a measure of nonlinearity along the (j, j') edge as

$$\vartheta_t(j, j') = \max_{0 \leq o \leq u} |\chi(o) - \pi(o)|.$$

(The tolerance $\tilde{\delta}$ for the LP duality gap may play a role in the above measure. However, if we have $\tilde{\delta} \ll \delta$, then the effect of the cell structure will be dominant.)

In case of the function $G(\mathbf{x})$, we consider the optimal objective value of the problem $P^\sharp(\check{\mathbf{x}}, \boldsymbol{\eta})$ as a function of $\boldsymbol{\eta}$, and define the nonlinearity measure

$\vartheta_t^\sharp(J, J')$ in a way similar to the above. A common measure of nonlinearity, in the direction t , of the recourse and constraint function approximations will then be

$$\vartheta_t = \max_{(J, J')} \vartheta_t(J, J') + \vartheta_t^\sharp(J, J'),$$

where maximization is considered for edges parallel to e_t . We cut the cell \check{C} along the coordinate $1 \leq t \leq \tilde{r}$ for which ϑ_t is maximal. Our cut is positioned in the barycenter $\underline{\eta}_{\check{C}}$. Neither of the new cells will be empty unless all the realizations $\eta_\omega \in \check{C}$ lie on the same hyperplane. In this case, however, the coordinate t would not have been selected for cutting.

The approximating distributions $\underline{\eta}$ and $\overline{\eta}$ are then refined in accordance with the cell structure. This procedure is iterated until δ -supporting functions are found.

3.2 Solution of the linear programming subproblems

We must solve many similar subproblems with increasing accuracy. We can use either simplex-type or interior-point methods.

The subproblems $P(\mathbf{x}, \eta)$ differ only in the right-hand-side vectors. If the representation (22) holds, then the possible right-hand-side vectors all come from a low-dimensional affine space. Similarity of the subproblems can easily be exploited by simplex-type methods. Optimal basic solutions together with the corresponding bases can be stored and reused as starting points for later runs. This procedure is called *warm start*.

If the distribution approximation of the previous section is employed, then the optimal basis of the problem corresponding to the barycenter of a cell will make a good starting dual feasible basis for the problems corresponding to the vertices of this cell (i.e., we set $\theta = 0$ to find exact solutions; after solving the problem $P(\check{\mathbf{x}}, \underline{\eta}_{\check{C}})$ the optimal basis is saved, and the problems $P(\check{\mathbf{x}}, \overline{\eta}_{\check{C}})$ are solved by dual simplex method starting from the saved basis). A similar scheme can be used for the problems P^\sharp . More sophisticated schemes are described in Gassmann and Wallace (1996).

New methods and implementation strategies proposed by Maros (2003b) led to substantial improvement in the efficiency of the dual simplex method.

On the other hand, interior-point methods seem more convenient to finding approximate solutions of the subproblems. There are effective interior-point methods that generate pairs of solutions; one of the pair being primal feasible, the other dual feasible. The LP duality gap is decreasing at each step. A state-of-art survey of such methods can be found in Roos et al. (1997), where the following efficiency estimate is proven: Let $\tilde{\delta} = \theta\delta > 0$ be a prescribed accuracy. In order to decrease the LP duality gap below $\tilde{\delta}$, the *primal-dual logarithmic barrier method* and the *predictor-corrector method* require $2\sqrt{d} \ln v/\tilde{\delta}$ iterations, where d is the dimension of the problem (in our case $\max(r, s)$); and

v measures the LP duality gap at the starting point. This estimate implies that considerable effort can be spared by early termination.

There are infeasible versions of these methods, where primal and dual feasibility is expected to be attained together with optimality. In practice, however, infeasibility is usually reduced much earlier than optimality is attained. Hence early termination is possible with such methods, too. These infeasible methods have a very good reputation, supported by many computational studies, see e.g. Andersen and Andersen (2000), and Gondzio and Grothey (2003).

Remark 13 If Assumption 6 holds, then by solving a recourse problem of the type (4), we also solve the related recourse problem of the type (6). Hence in this case the estimation of the infeasibility measure requires but a negligible computational effort.

4 Implementation issues

Our test program package was written in C language. The input is compatible with the SMPS format though it relies on special naming conventions that simplify data handling in case of two-stage problems.

The structure of our Level Decomposition implementation is depicted in Figure 1. The solver consists of three components. The main module is an implementation of the Inexact Constrained Level Method. An approximation scheme provides recourse and constraint function information in the form of cuts of appropriate accuracy. The CPLEX callable library is used for the solution of the linear and quadratic subproblems arising from decomposition, and for the solution of the recourse problems constructed in the approximation scheme.

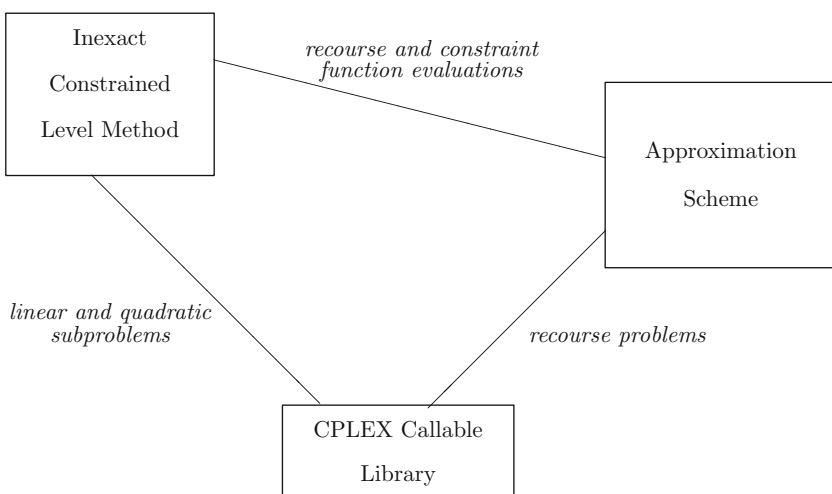


Fig. 1 The structure of our Level Decomposition implementation

Our implementation of the Inexact Constrained Level Method includes the bundle reduction technique mentioned in section 2, which means that inactive cuts are discarded after certain iterations. The dynamically changing bundle is stored in chained lists, where each element represents a single cut (the relevant δ -subgradient and function value are stored). Once a cut is discarded, the appropriate list element is unlinked from the active chain, and linked on the inactive chain.

In our implementation of the approximation scheme, the original distribution needs to be discrete. (We approximate a discrete distribution with coarser ones.) We are going to sketch the data structures using the notation of section 3.1. The cell structure is stored as a binary tree. Leaf elements of the tree represent cells. A cell cut is represented by linking two new leaf elements to the appropriate tree element (which ceases to be a leaf). Given a cell C , the following data are stored in the corresponding leaf element:

- the number N_C and the total weight p_C of those η -realizations that fall into this cell.
- coordinates of the barycenter $\underline{\eta}_C$.
- the gradient of the linear function $l_C(\mathbf{x})$, together with the lower approximation L_C of $F_C(\check{\mathbf{x}})$. The upper approximation U_C of $\bar{F}_C(\check{\mathbf{x}})$. (Here $\check{\mathbf{x}}$ denotes the current iterate of the master problem, in accordance with the notation of the previous section.)
- analogous data for the constraint function: the gradient of the linear function $l_C^\#(\mathbf{x})$, together with the lower approximation $L_C^\#$. The upper approximation $U_C^\#$.

The η -realizations are stored in a separate list. Those falling into a given cell are chained to the corresponding leaf element. We also maintain a list of the cell vertices. The vertices of a given cell are chained to the corresponding leaf element.

The data structures in our distribution approximation scheme are similar to those proposed by Mayer (1998). He implemented the L-Shaped Method imbedded into a classical approximation scheme.

We mention the technical detail that we do not approximate the distribution over cells having $N_C \ll 2^{\bar{r}}$. For such cells we make exact computation. (The rationale is that for such a cell, $\bar{\eta}$ would have more realizations than the original η .)

In the present implementation, the recourse problems are solved exactly (i.e., with high precision; with the notation of the previous section it means working with $\theta = 0$ and hence $\tilde{\delta} = 0$). The reason for not exploiting inexact solutions is that convergence of the CPLEX optimizer can be controlled through a *relative* tolerance, while our framework would require using an *absolute* tolerance. This is a purely technical problem that could be resolved by closer integration of the LP solver.

For the solution of the recourse problems we can presently apply either the CPLEX dual simplex optimizer or the CPLEX barrier optimizer. The CPLEX

dual simplex optimizer supports warm starts, while the CPLEX barrier optimizer presently does not.

5 Test problems and parameter setting

We tested with variants of the following two problems, taken from the *Portable Stochastic Programming Test Set* (POSTS) compiled by Derek Holmes. POSTS is available through the Stochastic Programming Community home page: <http://stoprog.org>.

PLTEXPA2 is the linear relaxation of a stochastic capacity expansion problem.

Problem sizes are: $m = 62$, $n = 188$, $r = 104$, $s = 272$.

Only the right-hand-side vector \mathbf{d} contains random parameters. In the original problem only seven components are random.

STORMG2 is based on a cargo flight scheduling application described by Mulvey and Ruszczyński (1995).

Problem sizes are: $m = 187$, $n = 121$, $r = 526$, $s = 1,259$.

Only the right-hand-side vector \mathbf{d} contains random parameters. In the original problem only 118 components are random.

We changed the objective function of the *PLTEXPA2* problem. Moreover we changed the distributions of the right-hand-side vectors in both test problems. The distributions we use have nothing in common with the original ones. In our variants, all components of the \mathbf{d} vectors are stochastic. In case of the *PLTEXPA2* problem, the 104-dimensional $\xi \equiv \mathbf{d}$ vector linearly depends on a 6-dimensional random η vector. In case of the *STORMG2* problem, there is no linear dependence among the 526 components of the \mathbf{d} vector. We created two sets of test problems:

PLTEXPA2, standard model. For the problems in this set, the S matrix and the realizations of the η vector were generated in such a manner that the first-stage problems (7) would be feasible. We sketch the construction: First a point $\hat{\mathbf{x}} \in X$ is selected. (Feasibility of the first-stage problem is ensured by making $\hat{\mathbf{x}}$ a feasible solution.) After this, \tilde{r} points are selected from the translated cone $T\hat{\mathbf{x}} + \{W\mathbf{y} \mid \mathbf{y} \geq \mathbf{0}\}$. Let these points be denoted by $\mathbf{s}_1, \dots, \mathbf{s}_{\tilde{r}}$. For $t = 1, \dots, \tilde{r}$, the ray $\{u\mathbf{s}_t \mid u \geq 0\}$ intersects the translated cone. Let $[\underline{\mathbf{s}}_t, \bar{\mathbf{s}}_t]$ denote the intersection interval. Realizations of η are generated so that $S\eta$ falls into the convex hull of the points $\underline{\mathbf{s}}_1, \bar{\mathbf{s}}_1, \dots, \underline{\mathbf{s}}_{\tilde{r}}, \bar{\mathbf{s}}_{\tilde{r}}$. (The S matrix is constructed by concatenating the columns $\mathbf{s}_1, \dots, \mathbf{s}_{\tilde{r}}$.)

We created six STOCH files containing 100,000, 200,000, ..., 600,000 scenarios, respectively.

The weight in the objective function was set to $w = 2,000$. This setting assures that Assumption 1 holds, hence we indeed solved a ‘traditional’ two-stage recourse problem. The setting $w = 2,000$ was found through trials with different settings. (*PLTEXPA2* is a capacity expansion problem, hence the recourse problems have a special combinatorial structure. In our opinion,

it is because of this special structure that Assumption 1 holds with such a moderately sized weight.)

STORMG2, relaxed model. The realizations of the η vector are samples from the 526-dimensional normal distribution having expectation vector $\mathbf{1} = (1, \dots, 1)^T$ and covariance matrix I . The norm $\|\cdot\|_\diamond = \|\cdot\|_1$ is used in the objective function with the weight $w = 2,000$. In the constraint $G(\mathbf{x}) \leq \Gamma$, we set $\Gamma = 2,000$. This is an arbitrary choice of parameters. We checked that Γ is large enough to make the relaxed problem (8) feasible. We also checked that Assumption 6 does not hold with the present setting of w . (This is not a ‘traditional’ two-stage recourse problem.)

We created twenty STOCH files containing 1000, 2000, ... 20,000 scenarios, respectively.

The CORE files of our variants of the test problems, and the STOCH files of our PLTEXPA2 standard model test set are available via the Internet at the address <http://www.cs.elte.hu/~fabian/testproblems>.

Test runs were made on a personal computer having 1,536 MHz CPU clock frequency and 1,033 Mb memory. We used Linux operating system. In this project our aim was building a test system in which different options are easily implemented and compared. Efficiency was considered of secondary importance.

In case of the PLTEXPA2-variants, we applied the distribution approximation scheme described in subsection 3.1. The STORMG2-variants were solved without approximation.

The starting iterate \mathbf{x}_1 was always selected by minimizing a randomly generated linear objective function over the feasible polyhedron X . The parameters controlling the underlying convex programming method were set to $\lambda = 0.5$, $\mu = 0.5$. In order to set the rest of parameters, we made some preliminary test runs. Let us sum up the results of these preliminary runs:

PLTEXPA2-variants. The original PLTEXPA2 problems have the peculiarity that the expected value of perfect information (EVPI) is very small. For PLTEXPA2_16, e.g., the optimal objective value is about -9.47935 and $\text{EVPI} \sim 10^{-6}$ holds.

Though we changed the objective function and the distribution, our PLTEXPA2-variants inherited the above peculiarity. For our problems, the optimal objective value is about $95,699.908$ and $\text{EVPI} \sim 0.64$ holds.

For the solution of these problems we set the starting accuracy to $\delta_1 = 1e6$. In order to see the influence of the parameter γ that controls the accuracy prescribed for the oracle, we solved the 600,000-scenario standard-model test problem with different settings. The problem was solved with eight-digit accuracy of the optimal objective value. The data in Table 1 are representative of our results. Larger γ values lead to more iterations, but less cells in the approximation scheme. Solution times show a strong correlation with the number of the LP optimizer calls. These quantities rise towards the endpoints of the examined interval. We decided use $\gamma = 0.15$ for the test runs.

Table 1 Results of preliminary test runs for tuning the run-time parameter γ that controls the accuracy prescribed for the oracle

γ	Iterations	Cells	1,000 calls	Solution time
0.01	>19	>12,451	>10,892	>2,000
0.03	22	11,984	9,574	1,669
0.06	25	11,637	9,500	1,608
0.09	28	11,203	9,925	1,664
0.12	30	10,652	9,070	1,396
0.15	33	10,352	9,078	1,348
0.18	37	10,284	9,854	1,467
0.21	42	10,213	11,010	1,658

The 600,000-scenario PLTEXPA2 standard model was solved with different settings. Stopping tolerance prescribed eight-digit accuracy each time. Row headers show the different γ settings. Columns contain: numbers of iterations, numbers of cells in the approximation scheme, numbers of CPLEX dual simplex optimizer calls in thousands, solution times in minutes. The > symbols in the first row indicate that we could not solve the problem with the setting $\gamma = 0.01$, due to memory shortage. Larger γ values lead to more iterations, but less cells in the approximation scheme

We found that the standard-model problems are close to having relatively complete recourse. Having started optimization from a point in $X \setminus K$, the feasible region $X \cap K$ was reached in a few master iterations, and infeasibility never occurred in later iterations.

STORMG2-variants. The optimal objective values of these problems were between 5,282,300 and 5,282,600. We found EVPI to be around 4,000.

For the solution of these problems we set the starting accuracy to $\delta_1 = 1e4$. The setting $\gamma = 0$ formally expresses that these problems were solved without approximation of the expected recourse function.

6 Test results

Let us sum up details of the tests that were run on the different problem sets:

PLTEXPA2, standard model. Problems from this set were solved with four different settings of the optimality tolerance: the solution process was terminated when the accuracy of the optimum reached five, six, seven, or eight digits, respectively.

Each problem was solved twice with each setting of the optimality tolerance; first by using the CPLEX dual simplex solver, and second time, by using the CPLEX barrier solver, for the solution of the recourse problems. When

using the dual simplex solver, we exploited the warm-start facility in a scheme described in section 3.2. When using the barrier solver, we could not exploit the similarity of the recourse problems.

Tables 2,3,4,5 and 6 summarize the results of those test runs where warm starts were exploited. In these tables, column headers show the prescribed accuracy in digits. In Tables 3,4,5 and 6, row headers show the numbers of the scenarios.

Test results of the runs not exploiting warm starts were very much similar to those cited in Tables 2,3 and 4. (Differences were under 10%.) In solution times, however, we found substantial differences: the barrier solver runs took 50–200% more time than the dual simplex solver runs.

STORMG2, relaxed model. These problems were solved with eight digit accuracy of the optimum. The recourse problems were solved by the CPLEX barrier solver. (There is no exploitable similarity in the recourse problems because the random vector is high dimensional and has independent components.)

Test results are summed up in Table 7.

Table 2 shows that in case of our PLTEXPA2 problem set, the Inexact Constrained Level Method converged at a steady rate: it required five or six master iterations to achieve an additional exact digit in optimum. However, the effort required for the expected recourse function evaluations grew rapidly with

Table 2 PLTEXPA2 standard model: Numbers of master iterations required to achieve different numbers of accurate digits in optimum

Accuracy in digits	5	6	7	8
Iterations	17	22	27 - 28	33

Iteration numbers proved virtually independent of the number of the scenarios. In case of 7-digit accuracy, the program made 27 iterations for some scenario numbers, and 28 iterations for others

Table 3 PLTEXPA2 standard model: Numbers of cells in the approximation scheme.

	5	6	7	8
100,000	110	1,259	1,943	2,205
200,000	30	1,591	3,131	4,080
300,000	27	1,211	3,491	5,096
400,000	26	816	5,033	7,074
500,000	26	523	4,404	7,983
600,000	27	810	6,126	10,352

Column headers show the prescribed accuracy in digits. Row headers show the numbers of the scenarios.

Table 4 PLTEXPA2 standard model: Numbers of CPLEX optimizer calls in thousands

	5	6	7	8
100,000	51	680	1,756	2,731
200,000	29	659	2,160	4,347
300,000	29	452	2,296	5,371
400,000	29	290	3,119	6,496
500,000	29	192	2,434	7,168
600,000	28	279	3,381	9,078

Calls required for the estimation of the expected recourse function and of the infeasibility measure are both counted

Table 5 PLTEXPA2 standard model: Solution times (in minutes) belonging to runs that exploit warm starts

	5	6	7	8
100,000	2	64	291	551
200,000	2	32	264	794
300,000	3	19	270	983
400,000	3	11	320	1,049
500,000	3	10	216	1,148
600,000	4	13	264	1,348

Recourse problems were solved by the CPLEX dual simplex solver

the accuracy prescribed. (The numbers of the cells in the approximation scheme and the numbers of the CPLEX optimizer calls are shown in Tables 3 and 4.) Solution times shown in Table 5 reflect the growing effort of approximation.

On the other hand, the effort of the function evaluations did not grow in proportion with the number of the scenarios. Let us consider the solution times in Table 5, and divide each item with the corresponding sample size given in the row header ; these ratios show a decreasing tendency.

Algorithmic structure of the Level Decomposition (LD) Method is such that the number of the LD master iterations required to solve a problem is independent of the number of the scenarios. Our test results seem to reinforce this theoretical observation.

For both of our test sets, the respective sets of the scenarios can be considered as random samples from the same continuous distribution. In this view,

Table 6 PLTEXPA2 standard model: Optimal objective values of the sample problems

	8
100,000	95,699.682
200,000	95,699.817
300,000	95,699.863
400,000	95,699.885
500,000	95,699.899
600,000	95,699.908

These data belong to solutions of eight-digit accuracy in optimum. Row headers show the numbers of the scenarios

we solved two different continuous problems by Monte Carlo sampling. Sampling of the random parameters is often used in practice, and the estimators of the optimal value and the optimal solution are well investigated. A state-of-art survey can be found in Shapiro (2003). We mention the following consistency result. Let O_N^* and S_N^* denote the optimal objective value, and the set of the optimal solutions, respectively, of the approximating problem constructed using a sample of size N . Let O^* and S^* denote the optimal objective value, and the set of the optimal solutions, respectively, of the true problem. Then

$$O_N^* \rightarrow O^* \quad (26)$$

and

$$D(S_N^*, S^*) := \sup_{\mathbf{x} \in S_N^*} \text{dist}(\mathbf{x}, S^*) \rightarrow 0 \quad (27)$$

with probability 1 as $N \rightarrow \infty$. (Here $\text{dist}(\mathbf{x}, S^*)$ means the distance from \mathbf{x} to S^* .)

The above consistency result also holds for the relaxed model (8) if the parameter Γ is properly set. (With large enough Γ , the Slater condition is satisfied. Together with the present assumptions, it implies consistency – see Remark 8 in Shapiro (2003).)

Our test results are in accordance with (26) and (27). Regarding our first test set, Table 6 shows optimal objective values of our discrete problems for different sample sizes. Regarding our second test set, optimal objective values are shown in the last column of Table 7. The sequence of the optimal objective values seems to converge in both cases, though the convergence is much slower for the relaxed model.

In case of the standard model, the sequence of the optimal objective values is noticeably monotonic. This characteristic is explained by the following inequality found by Norkin et al.(1998); and Mak et al. (1999):

Table 7 STORMG2 relaxed model.

	Iterations	Solution time	Optimum
1,000	23	42	5,282,318.0
2,000	26	93	5,282,339.8
3,000	22	119	5,282,351.4
4,000	26	187	5,282,304.0
5,000	23	207	5,282,329.8
6,000	26	282	5,282,341.7
7,000	22	277	5,282,380.4
8,000	22	316	5,282,380.6
9,000	27	436	5,282,413.7
10,000	30	540	5,282,424.2
11,000	26	515	5,282,427.9
12,000	23	498	5,282,429.6
13,000	23	538	5,282,427.2
14,000	22	553	5,282,443.4
15,000	22	595	5,282,459.7
16,000	22	634	5,282,431.9
17,000	23	706	5,282,444.7
18,000	23	747	5,282,451.2
19,000	25	853	5,282,431.7
20,000	21	755	5,282,424.9

Each of these problems was solved with eight-digit accuracy. Row headers show the numbers of the scenarios. Columns contain: numbers of master iterations, solution times in minutes, optimal objective values

$$\mathbb{IE} (O_N^*) \leq \mathbb{IE} (O_{N+1}^*) \leq O^*. \quad (28)$$

This inequality was originally proved for the relatively complete recourse case, but it readily generalizes for the standard (incomplete recourse) model (7). It may not hold for the relaxed model (8), though.

The eight-digit optimal solutions of our PLTEXPA2 problems have Euclidean norms of about 540, and they could be covered by a ball having a radius of

0.23. However, the spread is determined by a peculiar flatness of the objective function, rather than by sample sizes: We have solved the same 600,000-scenario problem with 8-digit accuracy several times, and found the spread of the minimizers to be in the magnitude of 1.

We mention that if the true problem has a single optimal solution and the ε -optimal set is small for small ε , then minimizers of the sample problems may converge very quickly, and the sequence may even stabilize for relatively small sample sizes. Linderoth et al. (2002) present extensive numerical experience regarding this phenomenon, and theoretical explanation was given by Shapiro and Homem-de-Mello (2000).

The 8-digit optimal solutions of our STORMG2 problems have Euclidean norms of about 535, and they could be covered by a ball having a radius of 0.68. However, these approximate optimal solutions lie along a straight line in the 121-dimensional space: projecting them onto a certain hyperplane we got points that could be covered by a ball having a radius of 0.08. This observation is in accordance with that of Linderoth et al. who observed that the ε -optimal set of the STORM problem is close to being a short line segment, rather than a large multidimensional basin. (STORM is similar to STORMG2.)

Remark 14 The eight-digit accuracy has little practical relevance: In general, probability distributions are coming via statistical estimates which are far less accurate. (In the present cases, the sample sizes allowed only five to six accurate digits.)

With setting such small stopping tolerances, our aim was only to explore the boundaries of the usability of the method. Let us mention that when we had prescribed an accuracy of no more than six digits, we observed quite reasonable solution times even with the present experimental version of the solver.

7 Conclusions and prospects

This paper contains the following new results:

- We examine conditions under which the expected recourse function can be extended to the whole space, and identify problem classes where this extension can be done in a computationally efficient way. Network recourse problems constitute such a problem class.

Such extension is also necessary in case of two-stage stochastic models using probabilistic constraints, see Prékopa (1995, chapter 12.10). Hence Proposition 3 and Corollary 9 has application in this field, too.

- By using the above extension and by explicitly accounting for infeasibility in the recourse subproblems, we formulate traditional two-stage stochastic programming problems as constrained convex programming problems. This formulation is called *standard model* in the paper.

This approach is practicable only if the expected recourse function can be extended in a computationally efficient way. In general, applicability critically depends on the magnitude of the penalty parameter w . Though Remark 5 justifies a heuristic procedure for setting the penalty parameter.

- We propose a new variant of the two-stage recourse model called *relaxed model* in the paper. This can be regarded as a generalization of the well-known *integrated chance constrained* model of Klein Haneveld (1986). We also give economic interpretations of the relaxed model. One of them substantially differs from traditional two-stage recourse models. It deals with the design and operation of a system that consumes certain resources in whose supply random interruptions may occur.
- For the above models, we propose a new solution method, called Level Decomposition in the paper. The method consists of two known components.
 - The first component is an approximative version of the Constrained Level Method of Lemaréchal, Nemirovskii, and Nesterov (1995). The approximative method was proposed by Fábíán (2000).
 - The second component involves the classic discretization and bounding methods worked out by Kall (1980), Kall and Stoyan (1982), and Birge and Wets (1986).

The novelty of Level Decomposition is the way these two components are integrated, and the fact that the master problem is solved by a convex programming method suited for constrained optimization.

In a typical implementation of the traditional solution framework, the main module is the distribution approximation scheme, and the submodule is the two-stage stochastic problem solver. To what extent the approximation should be refined between solver calls, is a difficult question, to which Mayer (1998) proposed a heuristic method. In Level Decomposition, the main module is the solver, and the approximating coarse distributions are refined in accordance with the solution process of the master problem. At each step, we have an estimate of how closely the optimum has been approached. At the beginning of the procedure, a rough approximation of the expected recourse function may be used. As the optimum is approached, the accuracy of the function evaluations can gradually be increased through gradual refinement of the approximating distributions. (In a single function evaluation, higher accuracy requires larger effort from the distribution approximation scheme. On the other hand, rough approximations may result in an increased number of iterations. Efficiency requires finding a balance between different efforts. Table 1 contains some experimental results concerning this issue: measures of various computational efforts are presented for different settings of the run-time parameter that controls accuracy prescribed for the approximation scheme.)

Using a convex programming method suited for constrained optimization means that feasibility and optimality issues are taken into consideration simultaneously. The bundle-type regularization extends to both feasibility and optimality issues. We avoid feasibility cuts that are constituents of the Benders family of solution methods. The price to be paid is that measuring infeasibility generally requires extra computational effort. Though Remark 13 shows that for certain problem classes, the extra effort is negligible.

We report encouraging test results with Level Decomposition. However, its utility can only be assayed by systematic comparisons with benchmark methods.

Simplex versus interior-point methods for the solution of the recourse problems. The cell structure of the distribution approximation offers natural means for batching recourse problems on the basis of similarity. Our test results show that the use of warm starts substantially decreases the time required for the solution of the recourse subproblems. It means that the dual simplex method is a good choice for the recourse problem solver. Efficiency can be enhanced by applying the special dual phase-II method of Maros (2003a).

Interior-point methods are reputed to have difficulties with warm start. However, Gondzio and Grothey (2003) proposed a reoptimization method to overcome these difficulties. They tested their method with an infeasible primal-dual solver, and report a substantial speedup. As a starting point, reoptimization needs a well-centered solution of the former problem. (An optimal solution of the former problem is not suitable, because it is close to the boundary of the feasible region.) This fact luckily coincides with our aim of finding near-optimal solutions. Measuring the utility of interior-point warm starts in Level Decomposition requires further investigation.

7.1 Possible fields of application

Sample average approximation. SAA frameworks are an alternative to discretization in case the representation (22) does not hold. In general, batches of sample problems are solved, and some optimality condition is tested by statistical tools. For example, Norkin et al. (1998) and Mak et al. (1999) construct confidence intervals for the optimality gap using the inequality (28). Besides estimating optimality gap, Shapiro and Homem-de-Mello (1998) propose another test based on the verification of the Karush–Kuhn–Tucker conditions.

Linderoth et al. (2002) used these statistical tools in estimating the optimum and in testing for (approximate) optimality. They solved two-stage stochastic programming problems with astronomical numbers of scenarios like 6×10^{81} . They produced good approximate solutions through sampling. (They used a trust-region method on a computational grid for the solution of their sample problems.)

If one uses Level Decomposition for the solution of sample problems in SAA frameworks, then the optimality tolerance can be tuned in accordance with sample sizes and with the number of the sample problems.

Models with probabilistic constraints. Prékopa (1973) proposed prescribing the feasibility of the recourse problem (1) by a probability $\geq p$ (with $0 < p < 1$, given parameter). In case the random vector has a logconcave distribution, logconcavity of the probabilistic constraint follows from the theory developed by Prékopa (1971). There are effective tools for the estimation of the probabilistic constraint function ; a state-of-art survey can be found in Prékopa

(2003). On the other hand, the expected recourse function can be estimated through discretization (or through sampling). The two types of estimation may be used in a common framework.

Models with Conditional-Value-at-Risk objectives and constraints. The CVaR risk measure was proposed by Rockafellar and Uryasev (2000). They derived a representation of CVaR that makes it tractable in optimization problems. An overview of CVaR-optimization models and methods can be found in Prékopa (2003) and in Kall and Mayer (2005).

Based on the polyhedral representation of Künzi-Bay and Mayer (2006), Fábíán (2005, 2006) proposed decomposition frameworks for handling CVaR objectives and constraints in two-stage stochastic models. Approximation of CVaR requires less effort than exact computation. This feature suggests that Level Decomposition may be useful for the solution of such problems.

Acknowledgments The authors would like to thank András Prékopa for suggesting this research and continuous encouragement, and Tamás Terlaky for his constructive remarks concerning the application of interior-point methods. We are grateful to István Maros and to the anonymous Referees for many valuable comments and suggestions that led to substantial improvement of the paper.

References

- Andersen ED, Andersen KD (2000) The MOSEK interior-point optimizer for linear programming: an implementation of the homogeneous algorithm. In: Frenk H, Roos K, Terlaky T, Zhang S, (eds) High performance optimization. Kluwer, Dordrecht, pp 197–232
- Beale EML (1955) On minimizing a convex function subject to linear inequalities. *J R Stat Soc Ser B* 17:173–184
- Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer Math* 4:238–252 republished in *Comput Manage Sci* 2:3–19 (2005)
- Birge JR, Louveaux FV (1997) *Introduction to Stochastic Programming*. Springer, Berlin Heidelberg New York
- Birge JR, Wets RJ-B (1986) Designing approximation schemes for stochastic optimization problems, in particular for stochastic programs with recourse. In: Prékopa A, Wets RJ-B (eds) *Stochastic Programming* 84, Vol 1. Mathematical Programming Study 27:54–102
- Björck Å (1996) *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics, Philadelphia
- Dantzig GB (1955) Linear programming under uncertainty. *Manage Sci* 1:197–206
- Dantzig GB, Madansky A (1961) On the solution of two-stage linear programs under uncertainty. In: *Proceedings of the 4th Berkeley symposium on mathematical statistics and probability*, vol 1: pp 165–176. University of California Press, Berkeley
- Dantzig GB, Wolfe P (1960) The decomposition principle for linear programs. *Oper Res* 8:101–111
- Fábíán CI (2000) Csentes T, Rapcsák T (eds) *Bundle-type methods for inexact data*. *Central Eur J Oper Res* 8 (special issue): 35–55
- Fábíán CI (2005) Decomposing CVaR minimization in two-stage stochastic models. *Stochastic Programming E-Print Series* 20
- Fábíán CI (2006) *Handling CVaR objectives and constraints in two-stage stochastic models*. RUTCOR Research Report, vol 5
- Frauendorfer K (1988) Solving SLP recourse problems with arbitrary multivariate distributions – the dependent case. *Math Oper Res* 13:377–394
- Frauendorfer K, Kall P (1988) A solution method for SLP recourse problems with arbitrary distributions – The independent case. *Probl Control Inf Theory* 17:177–205
- Gassmann HI, Wallace SW (1996) Solving linear programs with multiple right-hand sides: pricing and ordering schemes. *Ann Oper Res* 64:237–259

- Gondzio J, Grothey A (2003) Reoptimization with the primal-dual interior point method. *SIAM J Optim* 13:842–864
- Kall P (1980) Solving complete fixed recourse problems by successive discretization. In: Kall P, Prékopa A (eds) *Recent Results in Stochastic Programming*, Lecture Notes in Economics and Math. Systems 170. Springer, Berlin, Heidelberg New York, pp 135–138
- Kall P, Mayer J (2005) *Stochastic Linear Programming: Models, Theory, and Computation*. International series in operations research and management science. Springer, Berlin Heidelberg New York
- Kall P, Stoyan D (1982) Solving stochastic programming problems with recourse including error bounds. *Math Operforsch Stat Ser Optim* 13:431–447
- Kall P, Wallace SW (1994) *Stochastic programming*. Wiley, Chichester
- Kennington JL, Helgason RV (1980) *Algorithms for Network Programming*. Wiley, New York
- Klafszky E, Terlaky T (1992) On the ellipsoid method. *Radov Mat* 8:269–280
- Klein Haneveld WK (1986) Duality in stochastic linear and dynamic programming. *Lecture Notes in Economics and Mathematical Systems*, vol 274. Springer, Berlin Heidelberg New York
- Künzi-Bay A, Mayer J (2006) Computational aspects of minimizing conditional value-at-risk. *Comput Manage Sci* 3:3–27
- Lemaréchal C (1982) Basic theory in nondifferentiable optimization. Research Report No. 181, Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, Rocquencourt, France
- Lemaréchal C, Nemirovskii A, Nesterov Yu (1995) New variants of bundle methods. *Math Program* 69:111–147
- Linderoth JT, Shapiro A, Wright SJ (2002) The empirical behavior of sampling methods for stochastic programming. Optimization Technical Report 02-01. Computer Science Department, University of Wisconsin-Madison
- Mak W-K, Morton D, Wood RK (1999) Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Oper Res Lett* 24:47–56
- Maros I (2003a) A generalized dual phase-2 simplex algorithm. *Eur J Oper Res* 149:1–16
- Maros I (2003b) Computational techniques of the simplex method. Kluwer, Boston
- Mayer J (1998) *Stochastic linear programming algorithms*. Gordon and Breach, Amsterdam
- Mulvey JM, Ruszczyński A (1995) A new scenario decomposition method for large scale stochastic optimization. *Oper Res* 43:477–490
- Norkin VI, Pflug GCh, Ruszczyński A (1998) A branch and bound method for stochastic global optimization. *Math program* 83:425–450
- Prékopa A (1971) Logarithmic concave measures with applications to stochastic programming. *Acta Sci Math (Szeged)* 32:301–316
- Prékopa A (1973) Contributions to the theory of stochastic programming. *Math Program* 4:202–221
- Prékopa A (1995). *Stochastic Programming*. Kluwer, Dordrecht
- Prékopa A (2003) Probabilistic programming. In: Ruszczyński A, Shapiro A (eds) *Stochastic Programming*, Handbooks in Operations Research and Management Science vol 10, pp 267–351 Elsevier, Amsterdam
- Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. *J Risk* 2:21–41
- Roos C, Terlaky T, Vial J-Ph (1997) *Theory and Algorithms for Linear optimization*. Wiley, Chichester
- Ruszczyński A (1986) A regularized decomposition method for minimizing the sum of polyhedral functions. *Math Program* 35:309–333
- Ruszczyński A (2003) Decomposition methods. In: Ruszczyński A, Shapiro A (eds) *Stochastic Programming*. Handbooks in Operations Research and Management Science, vol 10, pp 141–211 Elsevier, Amsterdam
- Ruszczyński A, Shapiro A (2003) *Stochastic Programming Models*. In: Ruszczyński A, Shapiro A (eds) *Stochastic Programming*. Handbooks in Operations Research and Management Science, vol 10, pp 1–64 Elsevier, Amsterdam
- Ruszczyński A, Świętanowski A (1997) Accelerating the regularized decomposition method for two-stage stochastic linear problems. *Eur J Oper Res* 101:328–342
- Schrijver A (1986) *Theory of linear and integer programming*. Wiley, Chichester
- Shapiro A (2003) Monte Carlo sampling methods. In: Ruszczyński A, Shapiro A (eds) *Stochastic Programming*, Handbooks in Operations Research and Management Science, vol 10 pp 353–425 Elsevier, Amsterdam

- Shapiro A, Homem-de-Mello T (1998) A simulation-based approach to two-stage stochastic programming with recourse. *Math Program* 81:301–325
- Shapiro A, Homem-de-Mello T (2000) On the rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM J Optim* 11:70–86
- Szántai T (1988) A computer code for the solution of probabilistic-constrained stochastic programming problems. In: Ermoliev Yu, Wets RJ-B (eds) *Numerical Techniques for Stochastic Optimization*. Springer, Berlin Heidelberg New York
- Van Slyke R, Wets RJ-B (1969) L-Shaped linear programs with applications to optimal control and stochastic programming. *SIAM J Appl Math* 17:638–663
- Wets RJ-B (1974) Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM Review* 16:309–339
- Zakeri G, Philpott AB, Ryan DM (2000) Inexact cuts in Benders decomposition. *SIAM J Optim* 10:643–657