Computational aspects of risk-averse optimization in two-stage stochastic models

Csaba I. Fábián*

Abstract

Computational studies on two-stage stochastic programming problems indicate that aggregate models have better scale-up properties than disaggregate ones, though the threshold of breaking even may be high. In this paper we attempt to explain this phenomenon, and to lower this threshold.

We present the on-demand accuracy approach of Oliveira and Sagastizábal in a form which shows that this approach, when applied to two-stage stochastic programming problems, combines the advantages of the disaggregate and the aggregate models.

Moreover, we generalize the on-demand accuracy approach to constrained convex problems, and show how to apply it risk-averse two-stage stochastic programming problems.

Keywords. stochastic programming, risk-averse models, convex programming, cutting-plane methods, linear programming, simplex method.

1 Introduction

In this paper we argue for aggregate models in decomposition schemes for two-stage stochastic programming problems. We observe that analogous schemes proved effective for single-stage risk-averse problems, and for general linear programming problems.

A major drawback of the aggregate model for two-stage problems is that an aggregate master problem can not contain all the information obtained by the solution of the second-stage problems. Oliveira and Sagastizábal [33] develop special regularization methods for unconstrained convex optimization, namely, bundle level methods that use oracles with on-demand accuracy. The methods work with approximate function data, which is especially useful in solving stochastic problems. In order to apply the on-demand accuracy approach to two-stage stochastic programming problems, Oliveira and Sagastizábal propose inserting a new solver component between the aggregate master problem and the second-stage problems. The role of the new component is to provide approximate values and gradients of the expected recourse function, based on disaggregate information. In this paper we present the on-demand accuracy approach in a form which shows that it combines the advantages of the disaggregate and the aggregate approaches.

Moreover, we generalize the on-demand accuracy approach to constrained convex problems, adapting the constrained level method of Lemarechál, Nemirovskii, and Nesterov [24] to this framework. The new method can solve two-stage risk-averse problems. We show that aggregate models can be handled in a decomposition scheme practically without losing second-stage information.

The paper is organized as follows. In Section 2 we review some well-known means of risk aversion. In Section 3 we deal with two-stage stochastic programming problems. We discuss respective arguments for and against the disaggregate and the aggregate models, and present the on-demand accuracy approach. In Section 4, we deal with risk-averse two-stage stochastic programming problems, namely, a CVaR-constrained model of Ahmed [1], and the stochastic ordering-constrained model of Dentcheva and Martinez [7]. We show that such models can be handled by a constrained version of the on-demand accuracy approach.

*Department of Informatics, Kecskemét College, 10 Izsáki út, 6000 Kecskemét, Hungary. E-mail: fabian.csaba@gamf.kefo.hu.

2 Risk aversion in single-stage stochastic programming models

In this section we review some well-known means of risk aversion: expected shortfall, conditional value-at-risk, and second-order stochastic dominance.

2.1 Expected shortfall

Let R be a random variable representing uncertain yield. We assume that the expectation of R exists. Given $t \in \mathbb{R}$, let us consider $\mathrm{E}\left([t-R]_+\right)$, where $[.]_+$ denotes the positive part of a real number. This expression can be interpreted as expected shortfall with respect to the target t. Though we must mention that the term 'expected shortfall' is also used in a different meaning, especially in finance.

Assume that in a decision model, the random yield is a linear function of the decision vector $\boldsymbol{x}$, namely $R = R_{\boldsymbol{x}} = \boldsymbol{R}^T\boldsymbol{x}$, where $\boldsymbol{R}$ is a random vector of appropriate dimension and distribution. (A simple example is portfolio yield, where the components of $\boldsymbol{x}$ represent investments into different assets, and the components of $\boldsymbol{R}$ represent returns of the respective assets.) We can add a constraint limiting expected shortfall, in the form $\mathrm{E}\left([t-R_{\boldsymbol{x}}]_+\right) \leq \rho$ with a constant $\rho \in \mathbb{R}_+$. Constraints of this type were introduced by Klein Haneveld [21], under the name of integrated chance constraints. Let us assume that the random vector $\boldsymbol{R}$ has a discrete finite distribution. Let $\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(S)}$ denote the realizations, occurring with probabilities $p_1, \ldots, p_S$, respectively. Then the realizations of $R_{\boldsymbol{x}}$ will be $r_{\boldsymbol{x}}^{(s)} = \boldsymbol{r}^{(s)\,T}\boldsymbol{x}$ $(s = 1, \ldots, S)$.

An obvious way of constructing a linear programming representation of the integrated chance constraint $\mathrm{E}\left([t-R_{\boldsymbol{x}}]_+\right) \leq \rho$ is the following. For each $s = 1, \ldots, S$, we introduce a new variable to represent $[t-r_{\boldsymbol{x}}^{(s)}]_+$. We will call this lifting representation.

Klein Haneveld and Van der Vlerk [22] proposed the following polyhedral representation

$$\mathrm{E}\left([t-R_{\boldsymbol{x}}]_+\right) = \max_{J \subset \{1,\ldots,S\}} \sum_{s \in J} p_s \left(t - r_{\boldsymbol{x}}^{(s)}\right). \tag{1}$$

2.2 Conditional value-at-risk

The conditional value-at-risk (CVaR) measure was characterized by Rockafellar and Uryasev [36]-[37], and Pflug [35]. CVaR is now widely used in finance, and getting applied in other areas. A recent application is [30].

Let Q be a random variable representing uncertain cost or loss. We assume that the expectation of Q exists. Given a confidence level β $(0 < \beta \leq 1)$, the risk measure $\mathrm{CVaR}_\beta(Q)$ is the conditional expectation of the upper β-tail of Q. Rockafellar and Uryasev also establish the minimization rule

$$\mathrm{CVaR}_\beta(Q) = \min_{t \in \mathbb{R}} \left\{ t + \frac{1}{\beta}\mathrm{E}\left([Q-t]_+\right) \right\} \tag{2}$$

that facilitates the solution of optimization problems involving the CVaR risk measure. Assume that in a decision model, the random loss is a linear function of the decision vector $\boldsymbol{x}$, namely $Q = Q_{\boldsymbol{x}}$. Let us assume a discrete finite distribution, and let $q_{\boldsymbol{x}}^{(s)}$ $(s = 1, \ldots, S)$ denote the realizations of $Q_{\boldsymbol{x}}$, occurring with probabilities $p_1, \ldots, p_S$, respectively.

In order to construct a linear programming representation of the CVaR minimization, we can use the lifting representation. I.e., we can introduce a new variable for each positive part $[q_{\boldsymbol{x}}^{(s)} - t]_+$ $(s = 1, \ldots, S)$ in (2). An alternative, polyhedral, representation was proposed by Künzi-Bay and Mayer [23]:

$$\begin{array}{rl} \mathrm{CVaR}_\beta\left(Q_{\boldsymbol{x}}\right) = & \min\{t + \frac{1}{\beta}\vartheta\} \\ & \text{such that} \quad t, \vartheta \in \mathbb{R}, \\ & \sum\limits_{s \in J} p_s \left(q_{\boldsymbol{x}}^{(s)} - t\right) \leq \vartheta \quad \text{for each } J \subset \{1, \ldots, S\}. \end{array} \tag{3}$$

The representations (3) and (1) employ the same idea. Klein Haneveld and Van der Vlerk developed their representation earlier, but Künzi-Bay and Mayer obtained theirs independently, by interpreting the single-stage CVaR-minimization problem as a two-stage stochastic programming problem.

2.3 Stochastic dominance and a dominance measure

Stochastic dominance is a widely used tool in economy to compare uncertain prospects, see, e.g., [19]. In this paper we only deal with second-order stochastic dominance (SSD). Recent applications of SSD-based models are discussed in [11], [43].

Let R and R' be a random variables representing uncertain yields. We assume that the expectations of both variables exist. We say that R dominates R' with respect to second-order stochastic dominance, and use the notation $R \succeq_{SSD} R'$, if either of the following equivalent conditions hold:

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

(b) $\mathrm{E}\,([t-R]_+) \leq \mathrm{E}\,([t-R']_+)$ holds for each $t \in \mathbb{R}$.

(c) $\mathrm{Tail}_\beta(R) \geq \mathrm{Tail}_\beta(R')$ holds for each $0 < \beta \leq 1$, where $\mathrm{Tail}_\beta(R)$ denotes the unconditional expectation of the lower β-tail of R.

Concavity of the utility function in *(a)* characterizes risk averse behaviour. For the equivalence of *(a)* and *(b)*, see e.g. [44]. The equivalence of *(b)* and *(c)* is shown by Ogryczak and Ruszczyński [32]. The criterion *(c)* is related to conditional value-at-risk, as we have $\mathrm{Tail}_\beta(R) = -\beta\,\mathrm{CVaR}_\beta(Q)$ with the uncertain loss $Q = -R$. In general, SSD relations can be described with a continuum of constraints.

Dentcheva and Ruszczyński in [8] and [9] introduced SSD constraints in stochastic models and explored mathematical properties of the resulting optimization problems. Let us assume that in a decision model, the random yield is a linear function of the decision vector $\boldsymbol{x}$, namely $R = R_{\boldsymbol{x}}$. Assume moreover that a reference random return $\widehat{R}$, with a known (discrete) distribution, is available. Dentcheva and Ruszczyński introduced the constraint $R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R}$. They formulate the problem using the criterion *(b)* and prove that, in case of finite discrete distributions, the SSD relation can be characterized by a finite system of inequalities between expected shortfalls. For finite discrete distributions these authors also develop a duality theory in which dual objects are nondecreasing concave utility functions.

Roman, Darby-Dowman, and Mitra in [38] use criterion *(c)*. They assume finite discrete distributions with equally probable outcomes, and prove that, in this case, the SSD relation can be characterized by a finite system of inequalities. Namely, prescribing the tail inequalities for $\beta = \frac{s}{S}$ $(s = 1, \ldots, S)$ is sufficient. Based on this observation, they propose choosing $\boldsymbol{x}$ such that the return distribution $R_{\boldsymbol{x}}$ comes close to, or emulates, the reference return $\widehat{R}$ in a uniform sense. Uniformity is meant in terms of differences among tails; i.e., the 'worst' tail difference

$$\min_{1 \leq s \leq S} \left\{ \mathrm{Tail}_{\frac{i}{S}}\left(R_{\boldsymbol{x}}\right) - \mathrm{Tail}_{\frac{i}{S}}\left(\widehat{R}\right) \right\} \tag{4}$$

is maximized. This can be considered a multi-objective model whose Pareto optimal solutions are SSD-efficient portfolios. (The origin of this multi-objective formulation can be traced back to [31].)

A scaled version of the above approach was proposed in [14], in the following simple form. A new decision variable $\vartheta \in \mathbb{R}$ is introduced, representing a 'certain' (i.e., riskless) yield. (In the portfolio optimization example, this means holding an amount of cash.) The variable ϑ is then maximized under the constraint $R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R} + \vartheta$. (In the portfolio optimization example, the latter relation means that we prefer the return $R_{\boldsymbol{x}}$ to the combined return of the stock index and ϑ amount of cash.) A dominance measure is defined as

$$\Theta\left(R_{\boldsymbol{x}}\right) = \max\left\{ \vartheta \in \mathbb{R} \;\middle|\; R_{\boldsymbol{x}} \succeq_{SSD} \widehat{R} + \vartheta \right\}. \tag{5}$$

A dual characterization is also presented in [14], together with different representations and solution approaches.

The computational study [15] compares the dominance measures (4) and (5), applied in a simple decision model. The results show that (4) focuses on extreme tails, while (5) replicates the shape of the benchmark distribution. (I.e., let $\boldsymbol{x}^\star$ denote the optimal solution of the dominance maximization problem. The shape of the distribution function belonging to $R_{\boldsymbol{x}^\star}$ is similar to the shape of distribution function belonging to $\widehat{R}$.)

2.4 On the efficiency of the polyhedral representation

Computational studies comparing the lifting and the polyhedral representations discussed in this section, unanimously report that the latter type performs much better on large problem instances: [22], [23], [39], [13], [26], [27]. In this section we attempt to explain this remarkable efficiency of the polyhedral representation.

Section A of the Appendix deals with general linear programming problems and computing techniques. We consider respective analogues of the lifting and the polyhedral representations. Namely, an abstract analogue of the lifting representation is problem (32.D) that makes a primal-dual pair with (32.P). On the other hand, an abstract analogue of the polyhedral representation is problem (35.P). Aside from the individual upper bounds in (32.P), and a single generalized upper bound in (35.P), the columns of the latter problem are aggregates of those of the former.

In the appendix we present linear programming arguments for the fact that the solution of the aggregate problem (35.P) generally requires much less simplex iterations than the solution of the disaggregate problem (32.P). The main argument, translated to the cut terminology, is that aggregate cuts tend to be less steep. I.e., a vector representing an aggregate cut tends to have a norm smaller than a vector representing a disaggregate cut.

We note that the discussion in the appendix is restricted to a special case where the relationship can be formally demonstrated. Namely, to the case when the above mentioned individual upper bounds and the single generalized upper bound are all redundant. Though no formal proof is presented for the case when these bounds are non-redundant, the arguments indicate a similar relationship between the lifting and the polyhedral representations in general.

3 Two-stage stochastic programming models and methods

We are going to discuss the effect of aggregation in two-stage stochastic programming problems. The present notation emphasizes the analogy between the disaggregate two-stage stochastic programming problem formulation and the lifting representation of Section 2 on the one hand, and the analogy between the aggregate two-stage stochastic programming problem formulation and the polyhedral representation of Section 2 on the other hand. – This analogy was first observed by Künzi-Bay and Mayer [23], in case of simple recourse problems.

In Section 3.1 we discuss respective arguments for and against the disaggregate and the aggregate models. Then in Sections 3.2 and 3.3 we discuss the on-demand accuracy approach of Oliveira and Sagastizábal [33], in a form which shows that this approach combines the advantages of the disaggregate and the aggregate models.

In the present discussion we assume discrete finite distributions, and linear functions. Moreover, we consider only problems with a bounded feasible domain and relatively complete recourse.

The first-stage decision is represented by the vector $\boldsymbol{x}$. Assume there are S possible outcomes (*scenarios*) of the random event, the sth outcome occurring with probability p_s. Suppose the first-stage decision has been made with the result $\boldsymbol{x}$, and the sth scenario has realized. The second-stage decision $\boldsymbol{y}$ is computed by solving the following *second-stage problem* or *recourse problem* that we denote by $\mathcal{R}_s(\boldsymbol{x})$.

$$\begin{array}{rl} \min & \boldsymbol{q_s^T y} \\ \text{such that} & T_s\boldsymbol{x} + W_s\boldsymbol{y} = \boldsymbol{h_s}, \\ & \boldsymbol{y} \geq \boldsymbol{0}, \end{array} \tag{6}$$

where $\boldsymbol{q_s}$, $\boldsymbol{h_s}$ are given vectors and T_s, W_s are given matrices. Let K_s denote the set of those $\boldsymbol{x}$ vectors for which the recourse problem $\mathcal{R}_s(\boldsymbol{x})$ has a feasible solution. This is a convex polyhedron. For $\boldsymbol{x} \in K_s$, let $q_s(\boldsymbol{x})$ denote the optimal objective value of the recourse problem. We assume that $q_s(\boldsymbol{x}) > -\infty$. The polyhedral convex function $q_s : K_s \to \mathbb{R}$ is called the *recourse function.*

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

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \sum\limits_{s=1}^{S} p_s\, q_s(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in X, \\ & \boldsymbol{x} \in K_s \;\; (s = 1, \ldots, S), \end{array} \tag{7}$$

where $X := \{\boldsymbol{x} \,|A\boldsymbol{x} = \boldsymbol{b},\, \boldsymbol{x} \geq \boldsymbol{0}\}$ is a non-empty polyhedron describing the constraints, $\boldsymbol{c}$ and $\boldsymbol{b}$ are given vectors and A is a given matrix, with compatible sizes. The expectation part of the objective, $q(\boldsymbol{x}) = \sum_{s=1}^{S} p_s\, q_s(\boldsymbol{x})$, is called the *expected recourse function.* This is a polyhedral convex function with the domain $K := K_1 \cap \ldots \cap K_S$.

The two-stage stochastic programming problem (7) - (6) can be formulated as a single linear programming problem called the *deterministic equivalent problem.*

In this paper we assume that the feasible domain X is bounded, and that $X \subset K$, hence the constraints $\boldsymbol{x} \in K_s$ $(s = 1, \ldots, S)$ are redundant in (7). Let us denote the dual of $\mathcal{R}_s(\boldsymbol{x})$ by $\mathcal{D}_s(\boldsymbol{x})$:

$$\begin{array}{rl} \max & \boldsymbol{z^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \\ \text{such that} & W_s^T \boldsymbol{z} \leq \boldsymbol{q_s}, \end{array} \tag{8}$$

where $\boldsymbol{z}$ is a real-valued vector. The feasible region is a convex polyhedron that we assumed nonempty. Given $\boldsymbol{x} \in X$, the objective value is finite according to the assumption $X \subset K_s$.

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

$$\widetilde{q}_s(\boldsymbol{x}) := \max_{\boldsymbol{u_s} \in \widetilde{U}_s} \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \quad (\boldsymbol{x} \in X) \tag{9}$$

is a lower approximation of $q_s(\boldsymbol{x})$ over X. Having appropriate subsets $\widetilde{U}_s$ for $s = 1, \ldots, S$, the *disaggregate-form* cutting-plane approximation of the first-stage problem (7) is constructed as

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \sum\limits_{s=1}^{S} p_s \vartheta_s \\ \text{such that} & \boldsymbol{x} \in X, \quad \vartheta_s \in \mathbb{R} \;\; (s = 1, \ldots, S), \\ & \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \leq \vartheta_s \;\text{ holds for any } \boldsymbol{u_s} \in \widetilde{U}_s \quad (s = 1, \ldots, S). \end{array} \tag{10}$$

The expectation in the objective, $\widetilde{q}(\boldsymbol{x}) = \sum_{s=1}^{S} p_s\, \widetilde{q}_s(\boldsymbol{x})$, is called the disaggregate model function. This is a lower approximation of $q(\boldsymbol{x})$ based on the sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$.

An *aggregate form* of the first-stage problem (7) is

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \vartheta \\ \text{such that} & \boldsymbol{x} \in X, \quad \vartheta \in \mathbb{R}, \\ & \sum\limits_{s=1}^{S} p_s \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \leq \vartheta \;\text{ holds for any } (\boldsymbol{u_1}, \ldots, \boldsymbol{u_S}) \in \widetilde{\mathcal{U}}. \end{array} \tag{11}$$

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

$$f(\boldsymbol{x}) = \max_{(\boldsymbol{u_1}, \ldots, \boldsymbol{u_S}) \in \widetilde{\mathcal{U}}} \sum_{s=1}^{S} p_s \boldsymbol{u_s^T}(\boldsymbol{h_s} - T_s\boldsymbol{x}) \tag{12}$$

is called the aggregate model function. This is a lower approximation of the disaggregate model function $\widetilde{q}(\boldsymbol{x})$.

Remark 1 *By selecting basic solutions of the respective dual recourse problems into the sets* $\widetilde{U}_s$ $(s = 1, \ldots, S)$*, we can ensure that the model functions* $\widetilde{q}(\boldsymbol{x})$ *and* $f(\boldsymbol{x})$ *are Lipschitz continuous with a constant depending only on the data of the two-stage stochastic programming problem.*

3.1 Disaggregate vs. aggregate formulations of two-stage models: an overview

Cutting-plane methods were devised on the basis of either the disaggregate formulation (10) or the aggregate formulation (11).

The first solution method for two-stage stochastic programming problems was proposed by Dantzig and Madansky [5] who observed that the dual of the deterministic equivalent problem fits the prototype for the Dantzig-Wolfe decomposition [6]. This approach uses the disaggregate formulation. Van Slyke and Wets [42] proposed a cutting-plane approach based on the aggregate formulation. This approach turned out to be equivalent to the Benders decomposition [2] specially adapted to the deterministic equivalent problem. (Aggregation being the speciality of the adaptation.)

The regularized decomposition method of Ruszczyński [40] is a bundle-type method for the minimization of the sum of polyhedral convex functions over a convex polyhedron. Hence it is naturally applied to the disaggregate model. The box-constrained trust-region method of Linderoth and Wright [25] solves the disaggregate problem, using a special trust-region approach. Fábián and Szőke in [17] adapted the level method of Lemarechál, Nemirovskii, and Nesterov [24], and the inexact version [12], to the aggregate model. Oliveira, Sagastizábal and Scheimberg [34] proposed a special inexact bundle method for the solution of the aggregate model.

The difference between the aggregate and the disaggregate problem formulations may result in a substantial difference in the efficiency of the solution methods. By using disaggregate cuts, more detailed information is stored in the master problem. This is done at the expense of larger master problems. Based on the numerical results of [3] and [18], Birge and Louveaux [4] conclude that the multicut approach is in general more effective when the number of the scenarios is not significantly larger than the number of the constraints in the first-stage problem. Results of the computational study [46] confirm that the scale-up properties of solvers based on aggregate models are better than those of solvers based on disaggregate models, though the break-even thresholds are generally high. The results of the computational study [45] provide further insights into the effects of cut aggregation.

Trukhanov et al. [41] propose an adaptive aggregation method. The idea is to start with a low level of cut aggregation and increasing it in course of the solution process. A technique called cut consolidation is introduced by Wolf and Koberstein [45]. Cuts that have been inactive in course of the latest few iterations are discarded, but their aggregation is added to the master problem. Both papers report encouraging test results.

3.2 Applying an oracle with on-demand accuracy

Oliveira and Sagastizábal [33] develop special regularization methods for unconstrained convex optimization, namely, bundle level methods that use oracles with on-demand accuracy. The methods work with approximate function data, which is especially useful in solving stochastic problems. Approximate function values and subgradients are provided by an oracle with on-demand accuracy. The accuracy of the oracle is regulated by two parameters: the first is a descent target, and the second is a tolerance. If the estimated function value reaches the descent target, then the prescribed tolerance is observed. Otherwise the oracle just detects that the target can not be met, and returns rough estimations of the function data, disregarding the prescribed tolerance. The method is based on [24], [20], and [12]; and integrates the level-type and the proximal approach. [33] also contains a thorough computational study that demonstrates the effectiveness of the on-demand accuracy approach. In order to apply the on-demand accuracy approach to two-stage stochastic programming problems, Oliveira and Sagastizábal propose inserting a new solver component between the aggregate master problem and the second-stage problems. The role of the new component is

	objective function	current model function	current upper bound	current iterate	approx support to objective fct at current iterate
stoch.prg model	$\boldsymbol{c}^T\boldsymbol{x}+q(\boldsymbol{x})$	$\boldsymbol{c}^T\boldsymbol{x}+f(\boldsymbol{x})$	$\overline{D}$	$\widehat{\boldsymbol{x}}$	$\boldsymbol{c}^T\boldsymbol{x}+\widehat{l}(\boldsymbol{x})$
level method	$\varphi(\boldsymbol{x})$	$\varphi_i(\boldsymbol{x})$	$\overline{\phi}_i$	$\boldsymbol{x}_{i+1}$	$l(\boldsymbol{x})$

Figure 1: Assignment between the objects used in the aggregate two-stage stochastic programming model, and those used in the partly asymptotically exact version of the level method, described in Section B.1.2 of the Appendix. – We consider the objects as they are on the ith execution of step 3 of Algorithm 2, before incrementing the iteration counter.

to provide approximate values and gradients of the expected recourse function, based on the information represented in $\widetilde{U}_s$ $(s=1,\ldots,S)$.

We present the on-demand accuracy approach in a form which shows that it combines the advantages of the disaggregate and the aggregate approaches. Namely, let $\widehat{\boldsymbol{x}}$ denote the current iterate in a (regularized) cutting-plane method used to solve the master problem (11). If

$$\boldsymbol{c}^T\widehat{\boldsymbol{x}}+f(\widehat{\boldsymbol{x}}) \;\text{ is } \textit{significantly smaller} \text{ than } \; \boldsymbol{c}^T\widehat{\boldsymbol{x}}+\widetilde{q}(\widehat{\boldsymbol{x}}) \tag{13}$$

then an appropriate cut is added to the master problem practically without solving second-stage problems. As a rule, second-stage problems are solved only in case the above comparison does not hold. The aggregate model function value is easily evaluated as the sets $\widetilde{U}_s$ containing previously computed dual vectors are stored. In the remaining part of this section we show that (13) is a legitimate descent target rule according to the on-demand accuracy approach, assuming that the master problem is solved with a special level-type method. In Section B.1 of the Appendix, we describe such a special method, namely, the partly asymptotically exact level method. This method is applicable to the present problem: the feasible domain X is assumed bounded, and we can also ensure Lipschitz continuity according to Remark 1.

We are going to apply the on-demand accuracy approach to the aggregate problem. Let $\boldsymbol{x}^\star$ denote the best solution known at the present stage of the solution process. Let $\overline{D}$ denote our upper estimate of the objective value $\boldsymbol{c}^T\boldsymbol{x}^\star+q(\boldsymbol{x}^\star)$. The relation (13) will be formally expressed as

$$\boldsymbol{c}^T\widehat{\boldsymbol{x}}+\widetilde{q}(\widehat{\boldsymbol{x}}) \geq \kappa\Big\{\boldsymbol{c}^T\widehat{\boldsymbol{x}}+f(\widehat{\boldsymbol{x}})\Big\}+(1-\kappa)\overline{D}, \tag{14}$$

where κ is a constant, $0<\kappa<1$, set in accordance with the parameters of the underlying convex optimization method. If this inequality holds, then we update our aggregate model function by adding a linear support function of $\widetilde{q}(\boldsymbol{x})$ at $\widehat{\boldsymbol{x}}$. This support function is computed as

$$\widehat{l}(\boldsymbol{x})=\sum_{s=1}^{S} p_s\,\widehat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s-T_s\boldsymbol{x}), \tag{15}$$

where $\widehat{\boldsymbol{u}}_s\in\widetilde{U}_s$ are such that $\widetilde{q}_s(\widehat{\boldsymbol{x}})=\widehat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s-T_s\widehat{\boldsymbol{x}})$ $(s=1,\ldots,S)$. We are going to show that (14) assumed, the cut belonging to the above support function is, as a rule, legitimately added to the aggregate model function.

Figure 1 presents the assignment between the objects of the present stochastic programming model, and those of the level method. The relation (14) translated to the terms of Section B of the Appendix looks like this:

$$l(\boldsymbol{x}_{i+1}) \;\geq\; \kappa\varphi_i(\boldsymbol{x}_{i+1})+(1-\kappa)\overline{\phi}_i. \tag{16}$$

(16), above, is just criterion *(iii.)* of Corollary 8. – Criterion *(i.)* obviously holds, and *(ii.)* can be ensured according to Remark 1. – Hence Corollary 8 applies: $\widehat{l}(\boldsymbol{x})$ is legitimately added to the aggregate model function, and the second-stage problem need not be solved, assuming that $i > 1$ and the $(i-1)$th iteration is non-critical. But there are very few critical iterations, according to Remark 6. Even by solving the second-stage problems after every critical iteration, we would not exaggerate computational effort.

3.3 Efficiency considerations

In traditional implementations of the aggregate model, precious information is lost by cut aggregation. In order to generate a cut to the aggregate master problem, the appropriate second-stage problems are solved. Though the solution of S second-stage problems typically entails a major computational effort, only a single aggregate vector is stored. In this section we show that the method discussed in Section 3.2 can retrieve practically all the relevant information represented in the sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$, i.e., in the disaggregate model.

First we compare plain, unregularized solution methods for both the disaggregate and the aggregate model. Namely, the pure multicut method for the the disaggregate model on the one hand, and the example method of Section B.1.4 for the aggregate model on the other hand. Let us assume that the stopping tolerance ϵ is set to 0 in both methods.

Let us first consider the example method as applied to the aggregate model. Since κ is set to a value almost 1, the relation (14) takes the form $\boldsymbol{c}^T\widehat{\boldsymbol{x}} + \widetilde{q}(\widehat{\boldsymbol{x}}) > \boldsymbol{c}^T\widehat{\boldsymbol{x}} + f(\widehat{\boldsymbol{x}})$. Hence no second-stage problems are solved until $\widetilde{q}(\widehat{\boldsymbol{x}}) = f(\widehat{\boldsymbol{x}})$ holds with the current iterate $\widehat{\boldsymbol{x}}$.

Suppose now that $\widetilde{q}(\widehat{\boldsymbol{x}}) = f(\widehat{\boldsymbol{x}})$ holds with the current iterate $\widehat{\boldsymbol{x}}$ of the example method. Since $\widehat{\boldsymbol{x}}$ is a minimizer of the aggregate model function $f(\boldsymbol{x})$, and we have $\widetilde{q}(\boldsymbol{x}) \geq f(\boldsymbol{x})$ $(\boldsymbol{x} \in X)$, it follows that $\widehat{\boldsymbol{x}}$ is also a minimizer of the disaggregate model function $\widetilde{q}(\boldsymbol{x})$. For the sake of the simplicity of the discussion, let us assume that the minimizer of the disaggregate model function is unique in each step of the multicut method. We have just proven that the sequence of the iterates of the multicut method coincides with the sequence of the substantial iterates of the example method. It follows that in course of these two methods, the same set of second-stage problems are solved.

Of course this was just a theoretical comparison, regularized solution of the master problem is generally more effective. But according to the observations in Section B.1.4, the parameters of the partly asymptotically inexact level method can be set in such a manner that κ will be close to 1.

4 Risk-averse two-stage stochastic programming

In this section we generalize the on-demand accuracy approach to risk-averse two-stage stochastic programming problems. Ahmed [1], and more recently, Dentcheva and Martinez [7] proposed extending special risk constraints to two-stage problems. In this paper we assume that the feasible domain X is bounded. Given $\boldsymbol{x} \in X$, the recourse function values $q_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$ are considered as realizations of a random recourse function value $Q(\boldsymbol{x})$.

4.1 Mean-risk models

Ahmed [1] adds a risk constraint in the form

$$\mathcal{G}\,(\,Q(\boldsymbol{x})\,) \leq \rho \tag{17}$$

to the first-stage problem (7). The function $\mathcal{G}$ maps a certain family of random variables to the set of the real numbers, and ρ is a constant.

Ahmed identifies such $\mathcal{G}$ risk mappings which result a convex $\boldsymbol{x} \mapsto \mathcal{G}\,(\,Q(\boldsymbol{x})\,)$ function. Among others conditional value-at-risk (CVaR) is such a risk mapping. For the solution of the resulting problems, he develops special cutting plane methods. One type uses disaggregate cuts. The other type uses aggregate

cuts, and employs parametric programming to explore the efficient frontier. In this section we propose a solution method for the mean-CVaR problem that is based on aggregate models. To this end, we'll need a generalization of the on-demand accuracy approach of Oliveira and Sagastizábal to constrained convex problems.

The first-stage problem (7) in case of complete recourse, and using the notation $Q(\boldsymbol{x})$ for random value of the recourse function, takes the form

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \mathrm{E}\left(Q(\boldsymbol{x})\right) \\ \text{such that} & \boldsymbol{x} \in X, \\ & \mathrm{CVaR}_\beta\left(Q(\boldsymbol{x})\right) \leq \rho, \end{array} \tag{18}$$

where the parameter ρ is set by the decision maker. According to the computational formula (2), the CVaR constraint in the above problem can be formulated as

$$t + \frac{1}{\beta}\mathrm{E}\left(\left[Q(\boldsymbol{x}) - t\right]_+\right) \leq \rho, \tag{19}$$

introducing an extra variable $t \in \mathbb{R}$.

Assume that by repeated solution of the second-stage problems $\mathcal{R}_s$ $(s = 1, \ldots, S)$, we have obtained finite sets $\widetilde{U}_s$ of respective dual feasible solutions. These determine the respective approximate recourse functions $\widetilde{q}_s(\boldsymbol{x})$ as in (9). Given $\boldsymbol{x} \in X$, let us consider the function values $\widetilde{q}_s(\boldsymbol{x})$ $(s = 1, \ldots, S)$ as realizations of a random function value $\widetilde{Q}(\boldsymbol{x})$. Of course $\widetilde{q}(\boldsymbol{x}) = \mathrm{E}\big(\widetilde{Q}(\boldsymbol{x})\big)$ is a lower approximation of $q(\boldsymbol{x}) = \mathrm{E}\left(Q(\boldsymbol{x})\right)$.

In a similar manner, given $\boldsymbol{x} \in X$ and $t \in \mathbb{R}$, let us consider the values of the functions $\widetilde{r}_s(\boldsymbol{x}, t) = t + \frac{1}{\beta}\left[\widetilde{q}_s(\boldsymbol{x}) - t\right]_+$ $(s = 1, \ldots, S)$ as realizations of a random function value $\widetilde{R}(\boldsymbol{x}, t)$. Of course $\widetilde{r}(\boldsymbol{x}, t) = \mathrm{E}\big(\widetilde{R}(\boldsymbol{x}, t)\big) = t + \frac{1}{\beta}\mathrm{E}\big(\big[\widetilde{Q}(\boldsymbol{x}) - t\big]_+\big)$ is a lower approximation of the left-hand side of the CVaR constraint (19).

Clearly $\widetilde{q}(\boldsymbol{x})$ and $\widetilde{r}(\boldsymbol{x}, t)$ contain all the information represented by the dual feasible sets $\widetilde{U}_s$ $(s = 1, \ldots, S)$. The problem

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \widetilde{q}(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in X,\, t \in \mathbb{R} \\ & \widetilde{r}(\boldsymbol{x}, t) \leq \rho \end{array} \tag{20}$$

is analogue to the disaggregate formulation of the master problem.

An aggregate model contains only part of the information represented by the dual feasible sets. Assume we have $\widetilde{\mathcal{U}} \subset \widetilde{U}_1 \times \cdots \times \widetilde{U}_S$ and let us construct the aggregate model function $f(\boldsymbol{x})$ of $\mathrm{E}\left(Q(\boldsymbol{x})\right)$ as in (12). Moreover, let us construct an aggregate model function $g(\boldsymbol{x}, t)$ of the left-hand side of the CVaR constraint (19) in an analogous manner. This construction is also based on $\widetilde{\mathcal{U}}$, and cuts are generated according to the idea sketched in Section 2. (Details will be described below.) Hence our aggregate master problem will be

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + f(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in X,\, t \in \mathbb{R} \\ & g(\boldsymbol{x}, t) \leq \rho. \end{array} \tag{21}$$

We are going to use a partly asymptotically exact version of the constrained level method described in Section B.2.2 of the Appendix. This is a primal-dual method. Having set the dual variable $\widehat{\alpha}$ $(0 < \widehat{\alpha} < 1)$, (unconstrained) level method-type steps are applied to a composite objective that is a convex combination of the objective and the constraint function. The dual variable is kept unchanged as long as possible, hence the constrained method consists of runs of the unconstrained method. In our case, the current disaggregate model of the composite objective function is

$$\widehat{\alpha}\left(\boldsymbol{c^T x} + \widetilde{q}\left(\boldsymbol{x}\right)\right) + \left(1 - \widehat{\alpha}\right)\left(\widetilde{r}\left(\boldsymbol{x}, t\right) - \rho\right), \tag{22}$$

and the current aggregate model of the composite objective function is

$$\widehat{\alpha}\left(\boldsymbol{c^T x} + f(\boldsymbol{x})\right) + \left(1 - \widehat{\alpha}\right)\left(g(\boldsymbol{x}, t) - \rho\right). \tag{23}$$

	current model of objective function	current model of constraint function	current dual iterate	current model of composite function	best upper estimate for a comp fct value	current (primal) iterate	appr support to obj fct at current iterate	appr support to constr fct at current iterate
	$\boldsymbol{c}^T\boldsymbol{x} + f(\boldsymbol{x})$	$g(\boldsymbol{x}, t) - \rho$	$\widehat{\alpha}$	(23)	$\overline{D}$	$\big(\widehat{\boldsymbol{x}}, \widehat{t}\big)$	$\boldsymbol{c}^T\boldsymbol{x} + \widehat{l}(\boldsymbol{x})$	$\widehat{l'}(\boldsymbol{x}, t) - \rho$
	$\varphi_i(\boldsymbol{x})$	$\psi_i(\boldsymbol{x})$	α_i	$\vartheta_i(\boldsymbol{x})$	$\overline{\theta}_i$	$\boldsymbol{x}_{i+1}$	$l(\boldsymbol{x})$	$l'(\boldsymbol{x})$

Figure 2: Assignment between the objects used in the CVaR-constrained master problem (21), and those used in the partly asymptotically exact version of the constrained level method, described in Section B.2.2 of the Appendix. – We consider the objects as they are on the ith execution of step 4 of Algorithm 11, before incrementing the iteration counter.

We optimize using aggregate models, but may generate (aggregate) cuts based on disaggregate information. Let $\big(\widehat{\boldsymbol{x}}, \widehat{t}\big)$ denote the current iterate, and let us compare aggregate model function values with disaggregate ones. If

$$\big(23 \,:\, \boldsymbol{x} = \widehat{\boldsymbol{x}}, t = \widehat{t}\,\big) \quad \text{is } \textit{significantly smaller} \text{ than} \quad \big(22 \,:\, \boldsymbol{x} = \widehat{\boldsymbol{x}}, t = \widehat{t}\,\big) \tag{24}$$

then appropriate cuts are added to the master problem practically without solving second-stage problems. As a rule, second-stage problems are solved only in case the above comparison does not hold. In the remaining part of this section we show that (24) is a legitimate descent target rule according to the on-demand accuracy approach, assuming that the master problem is solved with a special constrained level-type method. In Section B.2 of the Appendix, we describe such a special method, namely, the partly asymptotically exact constrained level method.

We are going to apply the on-demand accuracy approach to the aggregate problem. Let $(\boldsymbol{x}^\star, t^\star)$ denote the iterate whose value, when substituted into the composite objective function, is estimated smallest. Let $\overline{D}$ denote our upper estimate of this composite objective value. The relation (24) is then formulated as

$$\big(22 \,:\, \boldsymbol{x} = \widehat{\boldsymbol{x}}, t = \widehat{t}\,\big) \;\geq\; \kappa\,\big(23 \,:\, \boldsymbol{x} = \widehat{\boldsymbol{x}}, t = \widehat{t}\,\big) + (1-\kappa)\overline{D}, \tag{25}$$

where κ is a constant, $0 < \kappa < 1$, set in accordance with the parameters of the underying constrained convex optimization method. If this inequality holds, then we simultaneously update our respective aggregate models of the objective function and the constraint function. To the former, we add a linear support function of the disaggregate model of the objective function. To the latter, we add a linear support function of the disaggregate model of the constraint function.

A linear support function $\widehat{l}(\boldsymbol{x})$ of $\widetilde{q}(\boldsymbol{x})$ at $\widehat{\boldsymbol{x}}$ can be computed according to (15), and adding the appropriate cut to the aggregate model $f(\boldsymbol{x})$ means adding the new element $(\widehat{\boldsymbol{u}}_1, \ldots, \widehat{\boldsymbol{u}}_S)$ to $\widetilde{\mathcal{U}}$. The same vectors $\widehat{\boldsymbol{u}}_1, \ldots, \widehat{\boldsymbol{u}}_S$ can be used to construct a linear support function $\widehat{l'}(\boldsymbol{x}, t)$ of $\widetilde{r}(\boldsymbol{x}, t)$ at $\big(\widehat{\boldsymbol{x}}, \widehat{t}\big)$ in the form

$$\widehat{l'}(\boldsymbol{x}, t) = t + \frac{1}{\beta}\sum_{s\in\widehat{J}} p_s\left(\widehat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\boldsymbol{x}) - t\right) \quad \text{with} \quad \widehat{J} = \left\{s \in \{1,\ldots,S\} \,\middle|\, \widehat{\boldsymbol{u}}_s^T(\boldsymbol{h}_s - T_s\widehat{\boldsymbol{x}}) - \widehat{t} > 0\right\}, \tag{26}$$

and we can add the appropriate cut to the aggregate model $g(\boldsymbol{x}, t)$. We are going to show that (25) assumed, the respective cuts belonging to the support functions $\widehat{l}(\boldsymbol{x})$ and $\widehat{l'}(\boldsymbol{x}, t)$ are, as a rule, legitimately added to our aggregate models.

Figure 2 presents the assignment between the objects of the present stochastic programming model, and those of the constrained level method. The relation (25) translated to the terms of Section B of the Appendix looks like this:

$$\alpha_i l(\boldsymbol{x}_{i+1}) + (1-\alpha_i)l'(\boldsymbol{x}_{i+1}) \;\geq\; \kappa\vartheta_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\theta}_i. \tag{27}$$

(27), above, is just criterion *(iii.)* of Corollary 13. – Criterion *(i.)* obviously holds, and *(ii.)* can be ensured according to Remark 1. – Hence Corollary 13 applies: $\widehat{l}(\boldsymbol{x})$ and $\widehat{l}'(\boldsymbol{x},t)$ are legitimately added to the aggregate objective and constraint model function, respectively, and the second-stage problem need not be solved, assuming that $i > 1$ and the $(i-1)$th iteration is non-critical. But there are very few critical iterations. Even by solving the second-stage problems after every critical iteration, we would not exaggerate computational effort.

Finally let us note that the constant κ can be close to 1 according to the partly asymptotically exact version of the level method. This indicates that the above two-stage method can retrieve practically all the relevant information represented in the disaggregate model.

4.2 A stochastic ordering-constrained two-stage model

Dentcheva and Martinez [7] propose adding the constraint

$$Q(\boldsymbol{x}) \preceq_{IC} \widehat{Q} \tag{28}$$

to the first-stage problem (7). Here $\widehat{Q}$ is a random variable, representing a benchmark cost or loss. The relation $\preceq_{IC}$ between random variables is the increasing convex order. This is analogous to the second-order stochastic dominance relation $\succeq_{SSD}$. Given appropriate random variables $Q, \widehat{Q}$ representing costs, we have $Q \preceq_{IC} \widehat{Q}$ if and only if $-Q \succeq_{SSD} -\widehat{Q}$ holds. Translating the characterization *(b)* of second-order stochastic dominance to the terms of the increasing convex order, we obtain: $Q \preceq_{IC} \widehat{Q}$ if and only if $\mathrm{E}([Q-t]_+) \leq \mathrm{E}\big(\big[\widehat{Q}-t\big]_+\big)$ holds for each $t \in \mathbb{R}$. The random variable $[Q-t]_+$ represents excess of loss with respect to the threshold t.

Assuming relatively complete recourse, the IC-constrained problem can be formulated as

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \mathrm{E}\,(Q(\boldsymbol{x})) \\ \text{such that} & \boldsymbol{x} \in X, \\ & Q(\boldsymbol{x}) \preceq_{IC} \widehat{Q}. \end{array} \tag{29}$$

Applying results of Ogryczak and Ruszczyński [32], and Dentcheva and Ruszczyński [10], Dentcheva and Martinez develop new characterizations of the increasing convex order. They also construct further finite linear models, based on the results of Dentcheva and Ruszczyński cited in Section 2.3. Dentcheva and Martinez develop special decomposition methods for the solution of the resulting problems: a multicut method, a quantile decomposition method, and a method they call excess function decomposition. The latter method uses aggregate cuts in the constraint, and disaggregate cuts in the objective. The authors implemented these methods and present encouraging test results.

As an alternative approach to excess function decomposition, we propose defining an IC-measure, analogous to the dominance measure Θ described in Section 2.3. Let

$$\mathcal{H}\,(Q(\boldsymbol{x})) = \min\left\{\xi \in \mathbb{R} \,\middle|\, Q(\boldsymbol{x}) \preceq_{IC} \widehat{Q} + \xi\right\}. \tag{30}$$

Here ξ is a certain (i.e., non random) loss. Clearly $Q(\boldsymbol{x}) \preceq_{IC} \widehat{Q}$ holds if and only if $\mathcal{H}\,(Q(\boldsymbol{x})) \leq 0$. Hence the IC-constrained problem (29) can be formulated as

$$\begin{array}{rl} \min & \boldsymbol{c^T x} + \mathrm{E}\,(Q(\boldsymbol{x})) \\ \text{such that} & \boldsymbol{x} \in X, \\ & \mathcal{H}\,(Q(\boldsymbol{x})) \leq 0. \end{array} \tag{31}$$

It is easily seen that $\mathcal{H}\,(Q(\boldsymbol{x}))$ is a convex function of $\boldsymbol{x}$. Moreover, considering $Q(\boldsymbol{x}) \preceq_{IC} \widehat{Q} + \xi$ represented as a finite system of linear inequalities, a supporting linear function to $\mathcal{H}\,(Q(\boldsymbol{x}))$ can be constructed for a given $\widehat{\boldsymbol{x}}$. This allows constructing a convex polyhedral model of the function $\mathcal{H}\,(Q(\boldsymbol{x}))$, to be used in the optimization process.

On the other hand, we can store the results of all the second-stage problems solved. Given $\widehat{\boldsymbol{x}}$, we can compute an estimate of $\mathcal{H}\left(Q(\widehat{\boldsymbol{x}})\right)$ taking into account all the second-stage information stored. This allows the application of the on-demand accuracy approach.

5 Conclusion and discussion

Decomposition is an effective and time-honoured solution scheme for two-stage stochastic programming problems. It can be interpreted as a cutting-plane scheme applied to the first-stage variables. Traditionally, there are two approaches: one can use a disaggregate or an aggregate model. In this paper we argue for aggregate models. We expand on an analogy between the two-stage aggregate approach and a special polyhedral approach that recently proved remarkably effective for single-stage risk-averse problems. (This analogy was pointed out by Künzi-Bay and Mayer [23] in case of simple recourse problems.) We also make an attempt to explain the success of this polyhedral approach. The main argument, translated to the cut terminology, is that aggregate cuts tend to be less steep. I.e., a vector representing an aggregate cut tends to have a norm smaller than a vector representing a disaggregate cut.

In a decomposition scheme for two-stage stochastic programming problems, the aggregate master problem can not contain all the information obtained by the solution of the second-stage problems. In a traditional implementation, a large part of this information is lost by aggregation, which is considered a major drawback of the aggregate approach. Oliveira and Sagastizábal [33] develop special regularization methods for unconstrained convex optimization, namely, bundle level methods that use oracles with on-demand accuracy. The methods work with approximate function data, which is especially useful in solving stochastic problems. In order to apply the on-demand accuracy approach to two-stage stochastic programming problems, Oliveira and Sagastizábal propose inserting a new solver component between the aggregate master problem and the second-stage problems. The role of the new component is to store second-stage information, and, based on this information, to provide approximate values and gradients of the expected recourse function. In this paper we present the on-demand accuracy approach in a form which shows that it combines the advantages of the disaggregate and the aggregate approaches. To this end, the master problem needs to be solved with a special level-type method. We describe such a special method in Section B.1 of the Appendix.

Moreover, we generalize the on-demand accuracy approach to risk-averse two-stage stochastic programming problems. We show that aggregate models in decomposition schemes can be handled practically without losing second-stage information. To this end, we develop a generalization of the partly asymptotically exact level method of Oliveira and Sagastizábal. The new method handles constraint functions in convex problems. It is based on the constrained level method of Lemarechál, Nemirovskii, and Nesterov [24], and an inexact version [12]. In this paper we adapt the method to a CVaR-constrained model of Ahmed [1]. and to the stochastic ordering-constrained model of Dentcheva and Martinez [7]. (The latter model is re-formulated using an appropriate constraint function.)

Acknowledgement

This project was partially supported by the Hungarian fund 'Social Renewal Operational Programme' ('Társadalmi Megújulás Operatív Program'), projects TÁMOP-4.2.2.C-11/1/KONV-2012-0004 and TÁMOP-4.2.2.C-11/1/KONV-2012-0012. These sources of support are gratefully acknowledged.

Appendix

A A linear programming detour

Let us consider a standard-form linear programming problem with individual upper bounds on the variables. For the sake of simplicity we assume that each variable has an upper bound of 1. The discussion is easily extended to arbitrary finite positive bounds. Our problem can be formulated as (32.P). This makes a primal-dual pair with (32.D).

$$\begin{array}{lr|l|lr} & \max\ \boldsymbol{c}^T\boldsymbol{x} & & & \min\ \boldsymbol{b}^T\boldsymbol{y}+\mathbf{1}^T\boldsymbol{z} \\ (32.\mathrm{P}) & A\boldsymbol{x}\le\boldsymbol{b} & & (32.\mathrm{D}) & A^T\boldsymbol{y}+I\boldsymbol{z}\ge\boldsymbol{c} \\ & I\boldsymbol{x}\le\mathbf{1} & & & \boldsymbol{y}\ge\mathbf{0},\ \boldsymbol{z}\ge\mathbf{0}, \\ & \boldsymbol{x}\ge\mathbf{0}, & & & \end{array} \tag{32}$$

where A is an $m\times n$ matrix, I is an $n\times n$ identity matrix, and the vectors $\boldsymbol{b}$, $\boldsymbol{c}$ and $\mathbf{1}=(1,\dots,1)^T$ have appropriate dimensions. We assume that both problems in (32) are feasible (and hence both have a finite optimum).

Having fixed $\boldsymbol{y}$ in (32.D), the components of the vector $\boldsymbol{z}$ can be minimized individually. Hence (32.D) is equivalent to

$$\min_{\boldsymbol{y}\ge\mathbf{0}}\left\{\ \boldsymbol{b}^T\boldsymbol{y}\ +\ \nu(\boldsymbol{y})\ \right\}\quad\text{with}\quad \nu(\boldsymbol{y}):=\sum_{j=1}^{n}\left[c_j-\boldsymbol{a}_j^T\boldsymbol{y}\right]_+, \tag{33}$$

where $\boldsymbol{a}_j$ denotes the jth column of A. – Here $\nu(\boldsymbol{y})$ represents the sum of the infeasibilities belonging to $\boldsymbol{y}$ in (32.D). The concept and the formulation are analogous to those of the 'expected shortfall' in stochastic programming, that we described in Section 2. (32.D) is a lifting representation, while a polyhedral representation can be formulated as

$$\nu(\boldsymbol{y})=\max_{J\in\mathcal{N}}\sum_{j\in J}\left(c_j-\boldsymbol{a}_j^T\boldsymbol{y}\right), \tag{34}$$

where $\mathcal{N}$ denotes the collection of all the subsets of $\{1,\dots,n\}$.

Problem (33) with the polyhedral formulation (34) of the objective function presents itself to be solved with a cutting-plane method. Such a cutting-plane method is equivalent to a special dual simplex method applied to (35.D), below:

$$\begin{array}{lr|l|lr} & \max\ \sum\limits_{J\in\mathcal{N}} c_{_J}x_{_J} & & & \\ & \sum\limits_{J\in\mathcal{N}} \boldsymbol{a}_{_J}x_{_J}\ \le\boldsymbol{b} & & & \min\ \boldsymbol{b}^T\boldsymbol{y}+\zeta \\ (35.\mathrm{P}) & & & (35.\mathrm{D}) & \boldsymbol{a}_j^T\boldsymbol{y}+\zeta\ge c_{_J}\quad (J\in\mathcal{N}) \\ & \sum\limits_{J\in\mathcal{N}} x_{_J}\ \le 1 & & & \\ & x_{_J}\ge 0\ \ (J\in\mathcal{N}), & & & \boldsymbol{y}\ge\mathbf{0},\ \zeta\ge 0, \end{array} \tag{35}$$

where $\boldsymbol{a}_{_J}:=\sum_{j\in J}\boldsymbol{a}_j$ and $c_{_J}:=\sum_{j\in J}c_j$. The 'primal' pair of (35.D) is in turn (35.P). This latter problem has a decision variable for every subset J of $\{1,\dots,n\}$. A dual simplex method applied to (35.D) can be interpreted as a simplex method applied to (35.P).

Summing up: a cutting-plane method applied to the polyhedral formulation of Problem (33) is equivalent to a special dual simplex method applied to (35.D), which is in turn equivalent to a special simplex method applied to (35.P). The speciality of this simplex method is the pricing rule which gives preference to columns that have already been basic. – The equivalence of these methods, and this special pricing rule are discussed in more detail in [16].

On the other hand, the lifting representation of Problem (33) is (32.D). A dual simplex method applied to (32.D) is in turn equivalent to a simplex method applied to Problem (32.P).

Problem (35.P) will be dubbed the *aggregate primal problem*, while Problem (32.P) will be dubbed the *disaggregate primal problem*. We are going to compare these problems. We are going to restrict the examination to a special case where the relationship can be formally demonstrated. Though we can not present a formal proof in general, the arguments indicate a similar relationship for general problems.

Let us assume that in the disaggregate primal problem (32.P), the individual upper bounds $I\boldsymbol{x} \leq \mathbf{1}$ are redundant. When solving this problem by a simplex method, the corresponding slack variables will always remain basic. Hence this procedure is equivalent to solving the problem

$$\max \boldsymbol{c}^T \boldsymbol{x} \quad \text{such that} \quad A\boldsymbol{x} \leq \boldsymbol{b}, \;\; \boldsymbol{x} \geq \mathbf{0}. \tag{36}$$

Let us also assume that in the aggregate primal problem (35.P), the normalization constraint $\sum_{J \in \mathcal{N}} x_{_J} \leq 1$ is redundant. When solving this problem by a simplex method, the corresponding slack variable will always remain basic. Hence this procedure is equivalent to solving the problem

$$\max \sum_{J \in \mathcal{N}} c_{_J} x_{_J} \quad \text{such that} \quad \sum_{J \in \mathcal{N}} \boldsymbol{a}_{_J} x_{_J} \leq \boldsymbol{b}, \;\; x_{_J} \geq 0 \; (J \in \mathcal{N}). \tag{37}$$

Suppose we solve (36) by a simplex method. This procedure can be reproduced in course of the solution of (37), by using only singleton J sets. But columns $\boldsymbol{a}_{_J}$ belonging to non-singleton sets J entering the basis may result longer simplex steps and more improvement in the objective value. Indeed, the reduced cost of $\boldsymbol{a}_{_J}$ is always the sum of the reduced costs of the vectors $\boldsymbol{a}_j$ $(j \in J)$. On the other hand, we have

$$\|\widetilde{\boldsymbol{a}}_{_J}\| \leq \sum_{j \in J} \|\widetilde{\boldsymbol{a}}_j\|, \tag{38}$$

where $\widetilde{\boldsymbol{a}} := B^{-1}\boldsymbol{a}$ denotes column $\boldsymbol{a}$ transformed to the current basis. Hence a steepest edge pricing rule (which is considered a major factor in the success of today's dual simplex implementations) tends to prefer column $\boldsymbol{a}_{_J}$ to $\boldsymbol{a}_j$ $(j \in J)$.

The aggregation idea has been used formerly in simplex implementations. In the nineteen eighties, Murty et al. [29] and Mitra et al. [28] proposed feasible direction methods in a simplex framework. Starting with a basic feasible solution, a 'profitable' direction is constructed using a weighted sum of certain nonbasic column vectors. (This is more general than (37) that uses just a sum of column vectors, all weights being 1.) Moving in this direction as far as possible while retaining feasibility, a boundary point is reached that is generally in the relative interior of a face of the feasible polyhedron. A basic feasible solution is then found whose objective value is not worse than that of the relative interior point – this process is called 'purification'. The above procedure was repeatedly performed, with encouraging results. Independently, the author of the present paper also worked out a similar method in the late eighties. Several aggregate pivot steps were performed at the beginning of the solution process. No purification was performed, because the simplex implementation allowed nonbasic variables to stay in the interiors of their respective feasible intervals. After the aggregate steps, the usual pivot steps were performed. The method proved useful, especially in the presence of initial degeneracy.

B Convex optimization using oracles with on-demand accuracy

In Section B.1 we sketch a special (non-proximal) form of the level bundle method with on-demand accuracy of Oliveira and Sagastizábal [33]. By applying this special method in a decomposition scheme for two-stage stochastic programming problems, the advantages of the disaggregate and the aggregate approaches are shown to be combined. Then in Section B.2 we develop a generalization of this method to constrained convex problems.

The discussion is based on the level method and the constrained level method of Lemaréchal, Nemirovski, and Nesterov [24], and on inexact versions described in [12].

B.1 Unconstrained convex optimization

Let us consider the problem

$$\begin{array}{ll} \min & \varphi(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in X, \end{array} \tag{39}$$

where $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a convex function, and $X \subset \mathbb{R}^n$ is a convex bounded polyhedron. We assume that φ is Lipschitz continuous over X with the constant Λ.

We describe the level method and sketch the convergence proofs as developed by [24]. Having generated the iterates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i \in X$, and using an oracle to return supporting linear functions $l_j(\boldsymbol{x})$ to $\varphi(\boldsymbol{x})$ at $\boldsymbol{x}_j$ $(j = 1, \ldots, i)$, respectively, the cutting-plane model function of φ is

$$\varphi_i(\boldsymbol{x}) = \max_{1 \le j \le i} l_j(\boldsymbol{x}) \quad (\boldsymbol{x} \in X).$$

Obviously φ_i is a piecewise linear and convex function that inherits Lipschitz continuity from φ. We have $\varphi_i(\boldsymbol{x}_j) = \varphi(\boldsymbol{x}_j)$ $(j \le i)$ and $\varphi_1 \le \varphi_2 \le \ldots \le \varphi_i \le \varphi$.

The cutting plane method computes the iterate $\boldsymbol{x}_{i+1}$ by minimising φ_i over X. The level method is a natural refinement. The best function value obtained until the ith step,

$$\overline{\phi}_i = \min_{1 \le j \le i} \varphi(\boldsymbol{x}_j) \tag{40}$$

is used as an upper bound for the optimum. A lower bound is the minimum of the ith model function:

$$\underline{\phi}_i = \min_{\boldsymbol{x} \in X} \varphi_i(\boldsymbol{x}).$$

The gap between the above bounds is $\Delta_i = \overline{\phi}_i - \underline{\phi}_i$. The sequence of the upper bounds $\overline{\phi}_i$ is monotone decreasing, and the sequence of the lower bounds $\underline{\phi}_i$ is monotone increasing. Hence the gap is tightening at each step.

Let $0 < \lambda < 1$ be some preset parameter. Consider the level set

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

The next iterate $\boldsymbol{x}_{i+1}$ is computed by projecting $\boldsymbol{x}_i$ onto the level set X_i. That is,

$$\boldsymbol{x}_{i+1} = \arg \min_{\boldsymbol{x} \in X_i} \operatorname{dist}(\boldsymbol{x}_i, \boldsymbol{x}),$$

where dist means the Euclidean distance. (Setting $\lambda = 0$ gives the cutting-plane method.)

Algorithm 2 *The level method.*

2.0 Initialization.

Set the stopping tolerance $\epsilon > 0$.

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

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

Let $i = 1$ (iteration counter).

2.1 Bundle update.

Let $l_i(\boldsymbol{x})$ be a supporting linear function to $\varphi(\boldsymbol{x})$ at $\boldsymbol{x}_i$.

Compute $\overline{\phi}_i$.

2.2 Near-optimality check.

Let $\underline{\phi}_i = \min_{\boldsymbol{x} \in X} \varphi_i(\boldsymbol{x})$, where $\varphi_i(\boldsymbol{x}) = \max_{1 \le j \le i} l_j(\boldsymbol{x}_j)$ is the current model function.

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

2.3 Finding a new iterate.

Let $\boldsymbol{x}_{i+1}$ be the projection of $\boldsymbol{x}_i$ onto $X_i = \left\{ \boldsymbol{x} \in X \,|\, \varphi_i(\boldsymbol{x}) \le \underline{\phi}_i + \lambda\Delta_i \right\}$.

Increment i, and repeat from step *2.1*.

The key to the convergence proof of the level method is

Proposition 3 *Let t and s be natural numbers, $t < s$, and assume that the following inequality holds*

$$(1-\lambda)\Delta_t \le \Delta_s. \tag{41}$$

Then the number of iterations performed while getting from $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ cannot be greater than $C\Delta_s^{-2}$, where C is a constant that depends only on the problem characteristics, and on the level parameter.

Proof. Consider the intervals $\left[\underline{\phi}_t, \overline{\phi}_t \right] \supseteq \left[\underline{\phi}_s, \overline{\phi}_s \right]$, the lengths of which are Δ_t and Δ_s, respectively. The point $\underline{\phi}_t + \lambda\Delta_t$ divides the first interval into two subintervals, the upper one having a length of $(1-\lambda)\Delta_t$. According to the assumption (41), the upper subinterval cannot contain $\left[\underline{\phi}_s, \overline{\phi}_s \right]$ in the interior. Hence we have $\underline{\phi}_s \le \underline{\phi}_t + \lambda\Delta_t$. Let us select a point $\boldsymbol{u}_s \in \{ \boldsymbol{x} \in X \,|\, \varphi_s(\boldsymbol{x}) \le \underline{\phi}_s \}$. (This level set is not empty due to the definition of $\underline{\phi}_s$.) We clearly have $\boldsymbol{u}_s \in X_t$. Moreover it can be shown in a similar manner that each level set X_i $(t \le i \le s)$ contains $\boldsymbol{u}_s$.

As stated in step *2.3*, the iterate $\boldsymbol{x}_{i+1}$ is computed as the projection of $\boldsymbol{x}_i$ onto the level set X_i. Hence the iterates are getting closer and closer to the point $\boldsymbol{u}_s$. From the properties of the projection, we get

$$\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{u}_s)^2 - \mathrm{dist}(\boldsymbol{x}_{i+1}, \boldsymbol{u}_s)^2 \;\ge\; \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})^2 \qquad (t \le i \le s-1). \tag{42}$$

The right-hand side is underestimated by $\left(\varphi_i(\boldsymbol{x}_i) - \varphi_i(\boldsymbol{x}_{i+1}) \right) / \Lambda$ since the function φ_i is Lipschitz continuous with the constant Λ. Obviously, we have

$$\varphi_i(\boldsymbol{x}_i) \;\ge\; \overline{\phi}_i \quad \text{and} \quad \varphi_i(\boldsymbol{x}_{i+1}) \;\le\; \underline{\phi}_i + \lambda\Delta_i \qquad (t \le i \le s-1). \tag{43}$$

From the above inequalities, we obtain

$$\varphi_i(\boldsymbol{x}_i) - \varphi_i(\boldsymbol{x}_{i+1}) \;\ge\; (\overline{\phi}_i - \underline{\phi}_i) - \lambda\Delta_i \;=\; (1-\lambda)\Delta_i \qquad (t \le i \le s-1).$$

It follows that $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})^2 \ge \Delta_i(1-\lambda)/\Lambda \ge \Delta_s(1-\lambda)/\Lambda$. This gives a lower bound on the left-hand side of (42). Moreover, D being the diameter of the feasible domain, we have $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{u}_s) \le D$. Hence the number of the steps performed while getting from $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ cannot be greater than stated in the Proposition, with $C = \left(\Lambda D/(1-\lambda)\right)^2$. □

The convergence proof concludes as follows. The sequence of the iterations is divided into finite subsequences of successive iterations. The subsequences are such that the gap does not decrease substantially in course of a subsequence – i.e., (41) holds for any t, s falling into the same subsequence – and each subsequence is of maximal length with respect to this feature. The iterations belonging to such a subsequence are called *non-critical*, and the iterations dividing such subsequences are called *critical*.

Let $\Delta^{(\ell)}$ denote the gap at the beginning of the ℓth subsequence. We have $(1-\lambda)\Delta^{(\ell)} > \Delta^{(\ell+1)}$ by definition, hence the sequence $\Delta^{(\ell)}$ $(\ell = 1, 2, \ldots)$ decreases by a geometric progression. Proposition 3 is then used to construct an upper bound on the number of the iterations to be performed before the stopping tolerance ϵ was reached.

Observation 4 *We have $\varphi_i(\boldsymbol{x}_{i+1}) = \underline{\phi}_i + \lambda\Delta_i$ with the new iterate $\boldsymbol{x}_{i+1}$ found in step 2.3.*

We show that the assumption $\varphi_i(\boldsymbol{x}_{i+1}) < \underline{\phi}_i + \lambda\Delta_i$ leads to a contradiction. Indeed, let us consider the line segment $[\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$. We have $[\boldsymbol{x}_i, \boldsymbol{x}_{i+1}] \subset X$ due to the convexity of X. Moreover,

$$\varphi_i(\boldsymbol{x}_i) \ge \overline{\phi}_i > \underline{\phi}_i + \lambda\Delta_i > \varphi_i(\boldsymbol{x}_{i+1}) \tag{44}$$

holds according to our assumption. The function $\varphi_i(\boldsymbol{x})$ being continuous, there exists $\widehat{\boldsymbol{x}} \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$ such that $\varphi_i(\widehat{\boldsymbol{x}}) = \underline{\phi}_i + \lambda\Delta_i$. Of course $\widehat{\boldsymbol{x}} \in X_i$ by definition, and obviously $\mathrm{dist}(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}) < \mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$, which contradicts the definition of $\boldsymbol{x}_{i+1}$.

B.1.1 Applying a descent target: a partly inexact version of the level method

In this section we consider a partly inexact version, where the objective function value is computed only if a descent target is met. An iterate meeting the descent target will be called *substantial*. If $\boldsymbol{x}_j$ is a substantial iterate then the oracle returns a subgradient: $l_j(\boldsymbol{x}_j) = \varphi(\boldsymbol{x}_j)$. Otherwise the oracle just detects that the target can not be met, and returns rough estimations of the function data.

Let $\mathcal{S}_i \subset \{1, \ldots, i\}$ denote the set of the indices belonging to substantial iterates that occurred up to the ith iteration. Obviously we have $\varphi_j(\boldsymbol{x}_j) = \varphi(\boldsymbol{x}_j)$ for any $j \in \mathcal{S}_i$. Instead of using (40), the best function value will be computed as

$$\overline{\phi}_i = \min_{j \in \mathcal{S}_i} \varphi(\boldsymbol{x}_j). \tag{45}$$

Let us initialize $\overline{\phi}_0 = +\infty$, $\mathcal{S}_0 = \emptyset$, and instead of step *2.1* of Algorithm 2, let us substitute the following

2.1′ Bundle update.
Let $l_i(\boldsymbol{x})$ be a linear function such that

(i.) $l_i(\boldsymbol{x}) \le \varphi(\boldsymbol{x}) \quad (\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l_i\| \le \Lambda$ holds with the gradient, and

(iii.)
either $l_i(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}_i)$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}$,
or $l_i(\boldsymbol{x}_i) \ge \overline{\phi}_{i-1}$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1}$.

Compute $\overline{\phi}_i$ using (45).

Let us observe that $\varphi_i(\boldsymbol{x}_i) \ge \overline{\phi}_i$ holds, irrespective of $\boldsymbol{x}_i$ being substantial or not. Hence Proposition 3 applies to this partly inexact form of the level method as well, with the proof unchanged.

B.1.2 A partly asymptotically exact version of the level method

As in the previous section, an iterate meeting the descent target will be called substantial, and $\mathcal{S}_i \subset \{1, \ldots, i\}$ will denote the set of the indices belonging to substantial iterates up to the ith iteration. We further relax the requirements on the linear functions by using δ-subgradients instead of subgradients. The accuracy prescribed for the jth iterate will be δ_j. Hence $l_j(\boldsymbol{x}_j) + \delta_j \ge \varphi(\boldsymbol{x}_j)$ will hold if $j \in \mathcal{S}_i$. The best function value will be computed as

$$\overline{\phi}_i = \min_{j \in \mathcal{S}_i} \{\, \varphi_j(\boldsymbol{x}_j) + \delta_j \,\}. \tag{46}$$

Estimating function values with a tolerance proportional to the current gap proves sufficient. For this purpose let us set a parameter γ such that $0 < \gamma < (1-\lambda)^2$, and let $\kappa = \frac{\gamma}{1-\lambda}$. Let us initialize $\overline{\phi}_0 = +\infty$, $\mathcal{S}_0 = \emptyset$, and instead of step *2.1′* of Algorithm 2, let us substitute the following

2.1″ Bundle update.
If $i = 1$ then let $\delta_i > 0$, arbitrary. Otherwise let $\delta_i = \gamma \Delta_{i-1}$.

Let $l_i(\boldsymbol{x})$ be a linear function such that

(i.) $l_i(\boldsymbol{x}) \le \varphi(\boldsymbol{x}) \quad (\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l_i\| \le \Lambda$, and

(iii.)
either $l_i(\boldsymbol{x}_i) + \delta_i \ge \varphi(\boldsymbol{x}_i)$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}$,
or $l_i(\boldsymbol{x}_i) + \delta_i \ge \overline{\phi}_{i-1}$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1}$.

Compute $\overline{\phi}_i$ according to (46).

Let us observe that $\varphi_i(\boldsymbol{x}_i) + \delta_i \ge \overline{\phi}_i$ holds, irrespective of $\boldsymbol{x}_i$ being substantial or not.

Proposition 5 *Let t and s be natural numbers, $t < s$, and assume that $(1-\lambda)\Delta_t \le \Delta_s$ holds.*

Then the number of iterations performed while getting from $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ cannot be greater than $1 + C'\Delta_s^{-2}$, where C' is a constant that depends only on the problem characteristics, and on the parameters of the method.

Proof. From the assumption it follows that $(1-\lambda)\Delta_{i-1} \le \Delta_i$ holds for $t+1 \le i \le s$. Hence we have

$$\delta_i = \gamma\Delta_{i-1} \le \frac{\gamma}{1-\lambda}\Delta_i \quad (t+1 \le i \le s). \tag{47}$$

The proof of Proposition 3 applies, with minor modifications. The inequalities (43) change to

$$\varphi_i(\boldsymbol{x}_i) \;\ge\; \overline{\phi}_i - \delta_i \quad \text{and} \quad \varphi_i(\boldsymbol{x}_{i+1}) \;\le\; \underline{\phi}_i + \lambda\Delta_i \qquad (t \le i \le s-1),$$

and from these we get

$$\varphi_i(\boldsymbol{x}_i) - \varphi_i(\boldsymbol{x}_{i+1}) \;\ge\; (\overline{\phi}_i - \underline{\phi}_i) - \lambda\Delta_i - \delta_i \;=\; (1-\lambda)\Delta_i - \delta_i \quad (t \le i \le s-1),$$

Using (47), we can continue

$$(1-\lambda)\Delta_i - \delta_i \;\ge\; \left(1 - \lambda - \frac{\gamma}{1-\lambda}\right)\Delta_i \quad (t+1 \le i \le s-1).$$

Here $\chi = 1 - \lambda - \frac{\gamma}{1-\lambda}$ is positive due to the setting $\gamma < (1-\lambda)^2$. It follows that $\operatorname{dist}(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})^2 \ge \chi\Delta_i/\Lambda \ge \chi\Delta_s/\Lambda$. The proof then can be completed along the lines of that of Proposition 3. – The iteration $t \to (t+1)$ needs to be counted separately, as (47) does not apply to this iteration. □

The convergence proof is analogous to that of the level method. By critical iterations, the sequence of the iterations is divided into finite subsequences of successive non-critical iterations. Let $\Delta^{(\ell)}$ denote the gap at the beginning of the ℓth subsequence. We have $(1-\lambda)\Delta^{(\ell)} > \Delta^{(\ell+1)}$ by definition, hence the sequence $\Delta^{(\ell)}$ $(\ell = 1, 2, \ldots)$ decreases by a geometric progression. Proposition 5 is then used to construct an upper bound on the number of the iterations to be performed before the stopping tolerance ϵ was reached.

Remark 6 *Since the sequence $\Delta^{(\ell)}$ $(\ell = 1, 2, \ldots)$ decreases by a geometric progression, it follows that the number of the critical iterations to be performed before the stopping tolerance ϵ was reached, is in the order of $\log\left(\frac{1}{\epsilon}\right)$.*

Observation 7 *Assuming $i > 1$ and the iteration $(i-1) \to i$ non-critical, we have $\varphi_i(\boldsymbol{x}_{i+1}) = \underline{\phi}_i + \lambda\Delta_i$ with the new iterate $\boldsymbol{x}_{i+1}$ found in step 2.3.*

This is the inexact analogue of Observation 4, and the proof is also analogous. We have

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

The second inequality is proven by the same arguments used in the proof of Proposition 5: we have $\delta_i < (1-\lambda)\Delta_i$, due to (47) and $\gamma < (1-\lambda)^2$.

We show that the assumption $\underline{\phi}_i + \lambda\Delta_i > \varphi_i(\boldsymbol{x}_{i+1})$ leads to a contradiction. Indeed, the function $\varphi_i(\boldsymbol{x})$ being continuous, there exists $\widehat{\boldsymbol{x}} \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$ such that $\varphi_i(\widehat{\boldsymbol{x}}) = \underline{\phi}_i + \lambda\Delta_i$. Obviously $\widehat{\boldsymbol{x}} \in X_i$ and $\operatorname{dist}(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}) < \operatorname{dist}(\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$, which contradicts the definition of $\boldsymbol{x}_{i+1}$.

Corollary 8 *Let $l(\boldsymbol{x})$ be a linear function such that*

(i.) $l(\boldsymbol{x}) \le \varphi(\boldsymbol{x})$ $(\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l\| \le \Lambda$ *holds with the gradient, and*

(iii.) $l(\boldsymbol{x}_{i+1}) \ge \kappa\varphi_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\phi}_i$ *holds with* $\kappa = \frac{\gamma}{1-\lambda}$.

Assuming $i > 1$ and the iteration $(i-1) \to i$ non-critical, $l(\boldsymbol{x})$ is legitimately assigned to be $l_{i+1}(\boldsymbol{x})$.

We show that under the above assumptions, $l(\boldsymbol{x}_{i+1}) + \delta_{i+1} \geq \overline{\phi}_i$ holds, and hence the criteria of *2.1″* are satisfied by $l_{i+1}(\boldsymbol{x}) = l(\boldsymbol{x})$, indicating that $\boldsymbol{x}_{i+1}$ is a non-substantial iterate. Indeed, due to Observation 7, the right-hand side of criterion *(iii.),* above, is

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

Substituting the value of κ, we get $\kappa(1-\lambda)\Delta_i \;= \gamma\Delta_i \;= \delta_{i+1}$, which concludes the proof.

B.1.3 A special form of the partly asymptotically exact level method

We derive a special from of the partly asymptotically exact method of Section B.1.2. – This method falls into the category PI2 according to the taxonomy of [33].

The speciality is that a tolerance in the construction of a supporting function as allowed only in case the descent target is not met, but exact supporting functions are constructed in the substantial iterates. Instead of *(iii.)* in step *2.1″* of Algorithm 2, let us substitute the following

(iii.′)
$$\begin{array}{lll} \text{either} & l_i(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}_i); & \text{in which case let } \mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}, \\ \text{or} & l_i(\boldsymbol{x}_i) + \delta_i \geq \overline{\phi}_{i-1}; & \text{in which case let } \mathcal{S}_i = \mathcal{S}_{i-1}. \end{array}$$

Of course the arguments of Section B.1.2 hold with the special method also.

B.1.4 Parameter settings for the unconstrained methods

The level parameter λ must fall into the interval $(0, 1)$ in the methods discussed in Section B. Let us note that the setting $\lambda = 0$ in the level method results the plain, unregularized, cutting-plane method. – Of course the convergence proof of the level method does not apply with this extremal setting.

In the partly asymptotically exact versions of the level method, $0 < \gamma < (1-\lambda)^2$ must hold with the tolerance regulating parameter γ. Let us note that the parameter $\kappa = \frac{\gamma}{1-\lambda}$ in Corollary 8 can be close to 1 according to this method. Namely, any value from interval $(0, 1-\lambda)$ is a feasible choice for κ. – Indeed, having such κ, the setting $\gamma := \kappa(1-\lambda)$ will satisfy $0 < \gamma < (1-\lambda)^2$, as required.

Example: the special method of Section B.1.3 with extremal parameter setting. This will be used as a reference method among two-stage stochastic programming methods.

Let us set λ to 0, and κ to a value of almost 1. Then $\gamma := \kappa(1-\lambda)$ will also be very close to 1, and the tolerance δ_i will be very close to the gap $\overline{\phi}_{i-1} - \underline{\phi}_{i-1}$ for $i > 1$. Hence criterion *(iii.′)* in step *2.1″* of Algorithm 2 can be simplified to

(iii.″)
$$\begin{array}{lll} \text{either} & l_i(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}_i); & \text{in which case let } \mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}, \\ \text{or} & l_i(\boldsymbol{x}_i) > \underline{\phi}_{i-1}; & \text{in which case let } \mathcal{S}_i = \mathcal{S}_{i-1}. \end{array}$$

(For $i = 1$, exact support function is required. Hence the first iterate will be substantial.)

Of course the convergence proof of the partly asymptotically exact level method does not apply with this extremal parameter setting. Even so, the method is usable in case $\varphi(\boldsymbol{x})$ is a polyhedral function, and the graph of each support function $l_i(\boldsymbol{x})$ contains a respective (potential) facet of the graph of $\varphi(\boldsymbol{x})$.

B.2 Constrained convex optimization

Let $X \subset \mathbb{R}^n$ be a bounded convex polyhedron with diameter D. Let φ and ψ be $X \to \mathbb{R}$ convex functions, both satisfying the Lipschitz condition with the constant Λ. The problem to be solved is

$$\begin{array}{rl} \min & \varphi(\boldsymbol{x}) \\ \text{such that} & \boldsymbol{x} \in X,\ \psi(\boldsymbol{x}) \leq 0. \end{array} \tag{48}$$

We assume that ψ takes positive values as well as 0. It follows that (48) is really a constrained problem and is consistent.

We describe the constrained level method and sketch the convergence proofs as developed in [24]. Together with a cutting-plane model of φ, we also build a cutting-plane model of ψ:

$$\psi_i(\boldsymbol{x}) = \max_{1 \le j \le i} \{\, l'_j(\boldsymbol{x}_j) \,\} \qquad (\boldsymbol{x} \in X)$$

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

Let Φ denote the optimal objective value of problem (48). If Φ is known in advance, then the quality of an approximate solution $\boldsymbol{x} \in X$ can be measured by $\varepsilon(\boldsymbol{x}) = \max\{\, \varphi(\boldsymbol{x}) - \Phi,\, \psi(\boldsymbol{x}) \,\}$. A lower approximation for Φ is

$$\underline{\Phi}_i = \min\{\, \varphi_i(\boldsymbol{x}) \mid \boldsymbol{x} \in X,\, \psi_i(\boldsymbol{x}) \le 0 \,\}, \tag{49}$$

for which $\underline{\Phi}_1 \le \ldots \le \underline{\Phi}_i \le \Phi$.

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

$$\boldsymbol{x}_i^\star := \sum_{j=1}^{i} \varrho_j \boldsymbol{x}_j. \tag{50}$$

The weights $\varrho_1, \ldots, \varrho_i$ will be determined through the solution of the following problem:

$$\begin{aligned} \min \quad & \max \left\{ \sum_{j=1}^{i} \varrho_j \varphi(\boldsymbol{x}_j) - \underline{\Phi}_i,\ \sum_{j=1}^{i} \varrho_j \psi(\boldsymbol{x}_j) \right\} \\ \text{such that} \quad & \varrho_j \ge 0 \ (j = 1, \ldots, i), \quad \sum_{j=1}^{i} \varrho_j = 1. \end{aligned} \tag{51}$$

Let H_i denote the optimal objective value of (51), and let $\boldsymbol{x}_i^\star$ in (50) be computed using the optimal solution of (51). Then obviously $\boldsymbol{x}_i^\star \in X$, and from the convexity of the functions φ and ψ, it follows that $\varepsilon(\boldsymbol{x}_i^\star) \le H_i$.

The linear programming dual of (51) can be written as $\max_{\alpha \in [0,1]} \; h_i(\alpha)$ with

$$h_i(\alpha) = \min_{1 \le j \le i} \{\, \alpha(\varphi(\boldsymbol{x}_j) - \underline{\Phi}_i) + (1 - \alpha)\psi(\boldsymbol{x}_j) \,\}. \tag{52}$$

Obviously h_i is a concave function, and $h_1 \ge h_2 \ge \ldots$.

The constrained level method is a primal-dual method. Having set the dual variable α_i $(0 < \alpha_i < 1)$, an unconstrained level method-type step is applied to a composite objective that is a convex combination of the objective and the constraint function. The dual variable is kept unchanged as long as possible, hence the constrained method consists of runs of the unconstrained method.

Algorithm 9 *The constrained level method.*

9.0 Initialization.

Set the stopping tolerance $\epsilon > 0$.

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

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

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

9.1 Bundle update.

Let $l_i(\boldsymbol{x})$ and $l'_i(\boldsymbol{x})$ be a supporting linear functions to $\varphi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$, respectively, at $\boldsymbol{x}_i$.

Define the model functions $\varphi_i(\boldsymbol{x})$ and $\psi_i(\boldsymbol{x})$, and compute $\underline{\Phi}_i$.

9.2 Near-optimality check.
Define the dual function $h_i(\alpha)$, and compute its maximum H_i.

If $H_i < \epsilon$, then near-optimal solution found, stop.

9.3 Tuning the dual variable.
Determine the interval $I_i = [\underline{\alpha}_i, \bar{\alpha}_i] \subseteq [0, 1]$ on which h_i takes non-negative values.

Compute α_i:

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

9.4 Finding a new primal iterate.
Define the level set

$$X_i := \left\{ \boldsymbol{x} \in X \,\middle|\, \alpha_i \varphi_i(\boldsymbol{x}) + (1 - \alpha_i)\psi_i(\boldsymbol{x}) \leq \alpha_i \underline{\Phi}_i + \lambda h_i(\alpha_i) \right\},$$

and let $\boldsymbol{x}_{i+1}$ be the projection of $\boldsymbol{x}_i$ onto X_i.

Increment i, and repeat from *9.1*

We give explanations and sketch proofs on the basis of [24].

Tuning the dual variable. The procedure uses a dual variable α tuned in such a manner that $h_i(\alpha_i)$ will always be 'sufficiently close' to H_i. The set

$$I_i = \{\ \alpha \in [0, 1]\ \mid h_i(\alpha) \geq 0\ \} \tag{53}$$

is of course an interval due to the concavity of h_i. This interval is not empty since we have $H_i > \epsilon$, otherwise the algorithm would have stopped already. Let the subinterval $\hat{I}_i \subset I_i$ be obtained by shrinking I_i: the center of $\hat{I}_i$ will be the same as the center of I_i, and for the lengths, $|\hat{I}_i| = (1 - \mu)|I_i|$ will hold with some preset parameter $0 < \mu < 1$. At the ends of I_i, we have sections of lengths of $\frac{\mu}{2}|I_i|$ not covered in $\hat{I}_i$. Owing to the concavity of h_i, it follows that

$$h_i(\alpha) \geq \frac{1}{2}\mu H_i \tag{54}$$

holds for any $\alpha \in \hat{I}$. The aim of the selection of the dual iterate is to leave the dual iterate unchanged as long as possible.

Finding a new primal iterate. The primal iterate $\boldsymbol{x}_{i+1}$ is selected by applying an unconstrained level method iteration to the composite objective function $\vartheta(\boldsymbol{x}) = \alpha_i \varphi(\boldsymbol{x}) + (1 - \alpha_i)\psi(\boldsymbol{x})$. Clearly $\vartheta_i(\boldsymbol{x}) = \alpha_i \varphi_i(\boldsymbol{x}) + (1 - \alpha_i)\psi_i(\boldsymbol{x})$ is an appropriate cutting-plane model of the composite objective. The *best function value*, i.e., the lowest function value taken among the known iterates is $\overline{\theta}_i = \min_{1 \leq j \leq i} \vartheta(\boldsymbol{x}_j) = \alpha_i \underline{\Phi}_i + h_i(\alpha_i)$. A *lower function level* is selected specially as $\underline{\theta}_i = \alpha_i \underline{\Phi}_i$. Concerning the gap $\overline{\theta}_i - \underline{\theta}_i$, it is easy to verify that the convergence proof of the level method is applicable as long as the following two requirements are satisfied in each iteration:

- $\underline{\theta}_i \leq \overline{\theta}_i$ should hold, and
- there should exist a feasible point whose function value is lower than or equal to $\underline{\theta}_i$.

In the present case, $\underline{\theta}_i = \alpha_i \underline{\Phi}_i \leq \alpha_i \underline{\Phi}_i + h_i(\alpha_i) = \overline{\theta}_i$ holds owing to the selection $\alpha_i \in I_i$. As for the latter criterion, a minimizer of the model problem (49) obviously satisfies that. The gap between $\overline{\theta}_i$ and $\underline{\theta}_i$ is $h_i(\alpha_i)$.

Convergence. As we have seen, the procedure consists of runs of the (unconstrained) level method. The length of such a run can be estimated just as sketched in the convergence proof of the level method. On the other hand, a bound can also be constructed on the number of the runs: Let $|I^{(\sigma)}|$ denote the length of the interval (53) at the beginning of the σth run. Then

$$|I^{(\sigma+1)}| \leq 1/(2-\mu)\ |I^{(\sigma)}| \tag{55}$$

holds due to the selection of the dual iterate. Indeed, the interval $I^{(\sigma+1)}$ – with the possible exception of a segment shorter then $\frac{1}{2}\mu|I^{(\sigma+1)}|$ – must be contained in one of the halves of $I^{(\sigma)}$, otherwise the dual iterate would not have changed.

Hence the length of the interval decreases by a geometric progression. Since I_i is the support of the function $h_i(\alpha)$, and H_i is the maximum of this function, the latter must decrease with the length of the support. Indeed, it is easily seen that the function $h_i(\alpha)$ is Lipschitz continuous with the constant $2D\Lambda$. Moreover, if we are in the σth run of the unconstrained level method, and $\sigma > 1$, then due to (55), there exists $\alpha^0 \in [0,1]$ such that $h_i(\alpha^0) = 0$.

Observation 10 *We have $\vartheta_i(\boldsymbol{x}_{i+1}) = (1-\lambda)\underline{\theta}_i + \lambda\overline{\theta}_i$ with the new iterate $\boldsymbol{x}_{i+1}$ found in step 9.4.*

This follows from Observation 4: let us consider the level method step applied to the composite function $\vartheta(\boldsymbol{x})$.

B.2.1 Applying a descent target: a partly inexact version of the constrained level method

In this section we consider a partly inexact version, where exact function values are computed only in certain iterates, called substantial iterates. – An iterate is labeled substantial if it meets a certain descent target, formulated in terms of the composite objective $\vartheta(\boldsymbol{x})$.

Let $\mathcal{S}_i \subset \{1,\ldots,i\}$ denote the set of the indices belonging to substantial iterates up to the ith iteration. The best point after iteration i will be constructed in the form of a convex combination of the former substantial iterates:

$$\boldsymbol{x}_i^\star := \sum_{j\in\mathcal{S}_i} \varrho_j \boldsymbol{x}_j. \tag{56}$$

The weights $\varrho_1,\ldots,\varrho_i$ will be determined through the solution of the following problem:

$$\begin{aligned} \min \quad & \max\left\{ \sum_{j\in\mathcal{S}_i} \varrho_j\varphi(\boldsymbol{x}_j) - \underline{\Phi}_i,\ \sum_{j\in\mathcal{S}_i} \varrho_j\psi(\boldsymbol{x}_j) \right\} \\ \text{such that} \quad & \varrho_j \geq 0\ (j\in\mathcal{S}_i), \quad \sum_{j\in\mathcal{S}_i}\varrho_j = 1. \end{aligned} \tag{57}$$

Let H_i denote the optimal objective value of (57), and let $\boldsymbol{x}_i^\star$ in (56) be computed using the optimal solution of (57). Then obviously $\boldsymbol{x}_i^\star \in X$, and from the convexity of the functions φ and ψ, it follows that $\varepsilon(\boldsymbol{x}_i^\star) \leq H_i$.

The linear programming dual of (57) can be written as $\max_{\alpha\in[0,1]}\ h_i(\alpha)$ with

$$h_i(\alpha) = \min_{j\in\mathcal{S}_i}\ \{\ \alpha(\varphi(\boldsymbol{x}_j) - \underline{\Phi}_i) + (1-\alpha)\psi(\boldsymbol{x}_j)\ \}. \tag{58}$$

Algorithm 11 *A partly inexact version of the constrained level method.*

11.0 Initialization.

Set the stopping tolerance $\epsilon > 0$.

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

Set $\overline{\theta}_0 = +\infty$ (best funcion value in the level metod),
and $\alpha_0 = 0.5$ (dual variable, initial value indifferent).

Set $\mathcal{S}_0 = \emptyset$ (the set of the substantial indices).

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

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

11.1 Bundle update.

Let $l_i(\boldsymbol{x})$ and $l'_i(\boldsymbol{x})$ be linear functions such that

(i.) $l_i(\boldsymbol{x}) \leq \varphi(\boldsymbol{x})$, $l'_i(\boldsymbol{x}) \leq \psi(\boldsymbol{x})$ $(\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l_i\|, \|\nabla l'_i\| \leq \Lambda$, and

(iii.) either $\big(\, l_i(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}_i)$ and $l'_i(\boldsymbol{x}_i) = \psi(\boldsymbol{x}_i) \,\big)$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}$,
or $\alpha_{i-1} l_i(\boldsymbol{x}_i) + (1-\alpha_{i-1}) l'_i(\boldsymbol{x}_i) \geq \overline{\theta}_{i-1}$; in which case let $\mathcal{S}_i = \mathcal{S}_{i-1}$.

Define the model functions $\varphi_i(\boldsymbol{x})$ and $\psi_i(\boldsymbol{x})$, and compute $\underline{\Phi}_i$.

11.2 Near-optimality check.

Define the dual function $h_i(\alpha)$ – taking into account substantial iterates only –, and compute its maximum H_i.

If $H_i < \epsilon$, then near-optimal solution found, stop.

11.3 Tuning the dual variable.

Determine the interval $I_i = [\underline{\alpha}_i, \bar{\alpha}_i] \subseteq [0,1]$ on which h_i takes non-negative values.

Compute α_i:

– for $i = 1$, let $\alpha_1 := \frac{1}{2}(\bar{\alpha}_1 + \underline{\alpha}_1)$,

– for $i > 1$, let $\alpha_i := \begin{cases} \alpha_{i-1} & \text{if} \quad \underline{\alpha}_i + \frac{\mu}{2}|I_i| \leq \alpha_{i-1} \leq \bar{\alpha}_i - \frac{\mu}{2}|I_i|, \\ \frac{1}{2}(\bar{\alpha}_i + \underline{\alpha}_i) & \text{otherwise.} \end{cases}$

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

11.4 Finding a new primal iterate.

Define the level set

$$X_i := \Big\{ \boldsymbol{x} \in X \,\Big|\, \alpha_i \varphi_i(\boldsymbol{x}) + (1-\alpha_i)\psi_i(\boldsymbol{x}) \leq \alpha_i \underline{\Phi}_i + \lambda h_i(\alpha_i) \Big\}.$$

Let $\boldsymbol{x}_{i+1}$ be the projection of $\boldsymbol{x}_i$ onto X_i.

Increment i, and repeat from *11.1*

The procedure consists of runs of the partly inexact unconstrained level method applied to the composite objective function $\vartheta(\boldsymbol{x}) = \alpha_i \varphi(\boldsymbol{x}) + (1-\alpha_i)\psi(\boldsymbol{x})$. Clearly $\vartheta_i(\boldsymbol{x}) = \alpha_i \varphi_i(\boldsymbol{x}) + (1-\alpha_i)\psi_i(\boldsymbol{x})$ is an appropriate cutting-plane model of the composite objective. In accordance with (45) of the partly inexact level method in Section B.1.1, the best function value is $\overline{\theta}_i = \min_{j \in \mathcal{S}_i} \vartheta(\boldsymbol{x}_j) = \alpha_i \underline{\Phi}_i + h_i(\alpha_i)$. Let us observe that if $i > 1$ and $\alpha_i = \alpha_{i-1}$ then $\vartheta(\boldsymbol{x}_i) \geq \overline{\theta}_i$ holds, irrespective of $\boldsymbol{x}_i$ being substantial or not. (Indeed, if $i \notin \mathcal{S}_i$ and $\alpha_i = \alpha_{i-1}$ then we have $\overline{\theta}_i = \overline{\theta}_{i-1}$. Moreover, $\vartheta(\boldsymbol{x}_i) \geq \vartheta_i(\boldsymbol{x}_i) \geq \overline{\theta}_{i-1}$ follows from *(iii.)* in step *11.1*, taking into account $\alpha_i = \alpha_{i-1}$ again.)

A lower function level is selected specially as $\underline{\theta}_i = \alpha_i \underline{\Phi}_i$. It is easily seen that $\underline{\theta}_i \leq \overline{\theta}_i$ holds in each iteration, and that a feasible point whose function value is lower than or equal to $\underline{\theta}_i$ always exists. Hence the length of a partly inexact level method run can be estimated as in Section B.1.1. Consequently the convergence proof of the constrained level method applies to this partly inexact version as well.

B.2.2 A partly asymptotically exact version of the constrained level method

We further relax the requirements on the linear support functions by using δ-subgradients instead of subgradients. The accuracy in the jth iterate will be δ_j. If the iterate $\boldsymbol{x}_j$ is substantial then we'll have $l_j(\boldsymbol{x}_j) + \delta_j \geq \varphi(\boldsymbol{x}_j)$ and $l'_j(\boldsymbol{x}_j) + \delta_j \geq \psi(\boldsymbol{x}_j)$.

Let $\mathcal{S}_i \subset \{1, \ldots, i\}$ denote the set of the indices belonging to substantial iterates up to the ith iteration. The best point after iteration i will be constructed in the form $\boldsymbol{x}_i^\star := \sum_{j \in \mathcal{S}_i} \varrho_j \boldsymbol{x}_j$. The weights $\varrho_1, \ldots, \varrho_i$ will be determined through the solution of the following problem:

$$\begin{aligned} \min \quad & \max \left\{ \sum_{j \in \mathcal{S}_i} \varrho_j \left(\varphi_i(\boldsymbol{x}_j) + \delta_j\right) - \underline{\Phi}_i, \sum_{j \in \mathcal{S}_i} \varrho_j \left(\psi_i(\boldsymbol{x}_j) + \delta_j\right) \right\} \\ \text{such that} \quad & \varrho_j \geq 0 \ (j \in \mathcal{S}_i), \quad \sum_{j \in \mathcal{S}_i} \varrho_j = 1. \end{aligned} \tag{59}$$

Let H_i denote the optimal objective value of (59), and let $\boldsymbol{x}_i^\star$ be computed using the optimal solution of (59). Then obviously $\boldsymbol{x}_i^\star \in X$, and from the convexity of the functions φ and ψ, it follows that $\varepsilon(\boldsymbol{x}_i^\star) \leq H_i$.

The linear programming dual of (59) can be written as $\max_{\alpha \in [0,1]} \ h_i(\alpha)$ with

$$h_i(\alpha) = \min_{j \in \mathcal{S}_i} \ \left\{ \ \alpha(\varphi_i(\boldsymbol{x}_j) - \underline{\Phi}_i) + (1-\alpha)\psi_i(\boldsymbol{x}_j) \ + \delta_j \ \right\}. \tag{60}$$

The parameter δ that regulates the precision of the oracle will be tuned in accordance with the gap. For this purpose let us set a parameter γ such that $0 < \gamma < (1-\lambda)^2$. Instead of step *11.1* of Algorithm 11, let us substitute the following

11.1′ Bundle update.

If $i = 1$ then let $\delta_i > 0$, arbitrary. Otherwise let $\delta_i = \gamma h_{i-1}(\alpha_{i-1})$.

Let $l_i(\boldsymbol{x})$ and $l'_i(\boldsymbol{x})$ be linear functions such that

(i.) $l_i(\boldsymbol{x}) \leq \varphi(\boldsymbol{x})$, $l'_i(\boldsymbol{x}) \leq \psi(\boldsymbol{x})$ $(\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l_i\|, \|\nabla l'_i\| \leq \Lambda$, and

(iii.)

either	$\big(l_i(\boldsymbol{x}_i) + \delta_i \geq \varphi(\boldsymbol{x}_i)$ and $l'_i(\boldsymbol{x}_i) + \delta_i \geq \psi(\boldsymbol{x}_i) \big)$;	in which case let	$\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{i\}$,
or	$\alpha_{i-1} l_i(\boldsymbol{x}_i) + (1-\alpha_{i-1}) l'_i(\boldsymbol{x}_i) + \delta_i \geq \overline{\theta}_{i-1}$;	in which case let	$\mathcal{S}_i = \mathcal{S}_{i-1}$.

Define the model functions $\varphi_i(\boldsymbol{x})$ and $\psi_i(\boldsymbol{x})$, and compute $\underline{\Phi}_i$.

The procedure consists of runs of the partly asymptotically exact unconstrained level method applied to the composite objective function $\vartheta(\boldsymbol{x}) = \alpha_i \varphi(\boldsymbol{x}) + (1-\alpha_i)\psi(\boldsymbol{x})$. Clearly $\vartheta_i(\boldsymbol{x}) = \alpha_i \varphi_i(\boldsymbol{x}) + (1-\alpha_i)\psi_i(\boldsymbol{x})$ is an appropriate cutting-plane model of the composite objective. In accordance with (46) of the partly asymptotically exact level method in Section B.1.2, the best function value is $\overline{\theta}_i = \min_{j \in \mathcal{S}_i} \{\vartheta_j(\boldsymbol{x}_j) + \delta_j\} = \alpha_i \underline{\Phi}_i + h_i(\alpha_i)$. Let us observe that if $i > 1$ and $\alpha_i = \alpha_{i-1}$ then $\vartheta_i(\boldsymbol{x}_i) + \delta_i \geq \overline{\theta}_i$ holds, irrespective of $\boldsymbol{x}_i$ being substantial or not.

A lower function level is selected specially as $\underline{\theta}_i = \alpha_i \underline{\Phi}_i$. It is easily seen that $\underline{\theta}_i \leq \overline{\theta}_i$ holds in each iteration, and that a feasible point whose function value is lower than or equal to $\underline{\theta}_i$ always exists. Hence the length of a partly asymptotically exact level method run can be estimated as in Section B.1.2. Consequently the convergence proof of the constrained level method applies to this partly asymptotically exact version as well.

Observation 12 *Let $i > 1$ be such that the iteration $(i-1) \to i$ non-critical in the corresponding unconstrained run. Then we have $\vartheta_i(\boldsymbol{x}_{i+1}) = (1-\lambda)\underline{\theta}_i + \lambda \overline{\theta}_i$ with the new iterate $\boldsymbol{x}_{i+1}$ found in step 11.4.*

This follows from Observation 7: let us consider the level method step applied to the composite function $\vartheta(\boldsymbol{x})$.

Corollary 13 *Let $l(\boldsymbol{x})$ and $l'(\boldsymbol{x})$ be a linear functions such that*

(i.) $l(\boldsymbol{x}) \leq \varphi(\boldsymbol{x})$, $l'(\boldsymbol{x}) \leq \psi(\boldsymbol{x})$ $(\boldsymbol{x} \in X)$,

(ii.) $\|\nabla l\|, \|\nabla l'\| \leq \Lambda$, *and*

(iii.) $\alpha_i l(\boldsymbol{x}_{i+1}) + (1-\alpha_i) l'(\boldsymbol{x}_{i+1}) \geq \kappa \vartheta_i(\boldsymbol{x}_{i+1}) + (1-\kappa)\overline{\theta}_i$ *holds with* $\kappa = \frac{\gamma}{1-\lambda}$.

Assuming $i > 1$ and the iteration $(i-1) \to i$ non-critical in the corresponding unconstrained run, $l(\boldsymbol{x})$ is legitimately assigned to be $l_{i+1}(\boldsymbol{x})$, and $l'(\boldsymbol{x})$ is legitimately assigned to be $l'_{i+1}(\boldsymbol{x})$.

This is the constrained analogue of Corollary 8, and the proof is also analogous.

References

[1] Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming* **106**, 433-446.

[2] Benders, J.F. (1962). Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik* **4**, 238-252. Republished, *Computational Management Science* **2** (2005), 3-19.

[3] Birge, J.R. and F.V. Louveaux (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research* **34**, 384-392.

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

[5] Dantzig, G.B. and A. Madansky (1961). On the Solution of Two-Stage Linear Programs under Uncertainty. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability* Vol. 1, 165-176. University of California Press, Berkeley.

[6] Dantzig, G.B. and P. Wolfe (1960). The Decomposition Principle for Linear Programs. *Operations Research* **8**, 101-111.

[7] Dentcheva, D. and G. Martinez (2012). Two-stage stochastic optimization problems with stochastic ordering constraints on the recourse. *European Journal of Operational Research* **219**, 1-8.

[8] Dentcheva, D. and A. Ruszczyński (2003). Optimization with stochastic dominance constraints. *SIAM Journal on Optimization* **14**, 548-566.

[9] Dentcheva, D. and A. Ruszczyński (2006). Portfolio optimization with stochastic dominance constraints. *Journal of Banking & Finance* **30**, 433-451.

[10] Dentcheva, D. and A. Ruszczyński (2010). Inverse cutting plane methods for optimization problems with second-order stochastic dominance constraints. *Optimization* **59**, 323-338.

[11] Drapkin D., R. Gollmer, U. Gotzes, F. Neise, and R. Schultz (2011). Risk Management with Stochastic Dominance Models in Energy Systems with Dispersed Generation. In: *Stochastic Optimization Methods in Finance and Energy* (M.I. Bertocchi, G. Consigli, and M.A.H. Dempster, editors), 253-271. Volume 163 of the International Series in Operations Research & Management Science. Springer.

[12] Fábián, C.I. (2000). Bundle-Type Methods for Inexact Data. *Central European Journal of Operations Research* **8** (special issue, T. Csendes and T. Rapcsák, eds.), 35-55.

[13] Fábián, C.I., G. Mitra, and D. Roman (2011). Processing second-order stochastic dominance models using cutting-plane representations. *Mathematical Programming* **130**, 33-57.

[14] Fábián, C.I., G. Mitra, D. Roman, and V. Zverovich (2011). An enhanced model for portfolio choice with SSD criteria: a constructive approach. *Quantitative Finance* **11**, 1525-1534.

[15] Fábián, C.I., G. Mitra, D. Roman, V. Zverovich T. Vajnai, E. Csizmás, and O. Papp (2011). Portfolio choice models based on second-order stochastic dominance measures: an overview and a computational study. In: *Stochastic Optimization Methods in Finance and Energy* (M.I. Bertocchi, G. Consigli, and M.A.H. Dempster, editors), 441-469. Volume 163 of the International Series in Operations Research & Management Science. Springer.

[16] Fábián, C.I., O. Papp, and K. eretnek (2011). Implementing the simplex method as a cutting-plane method. *Optimization Online,* July 2011.

[17] Fábián, C.I. and Z. Szőke (2007). Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science* **4**, 313-353.

[18] Gassmann, H.I. (1990). MSLiP: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming* **47**, 407-423.

[19] Hadar, J. and W. Russel (1969). Rules for ordering uncertain prospects. *The American Economic Review* **59** 25-34.

[20] Kiwiel, K.C. (1995). Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Mathematical Programming* **69**, 89-109.

[21] Klein Haneveld, W.K. (1986). Duality in Stochastic Linear and Dynamic Programming. *Lecture Notes in Economics and Math. Systems* **274** Springer-Verlag, New York.

[22] Klein Haneveld, W.K. and M.H. van der Vlerk (2006). Integrated chance constraints: reduced forms and an algorithm. *Computational Management Science* **3**, 245-269.

[23] Künzi-Bay, A. and J. Mayer (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science* **3**, 3-27.

[24] Lemaréchal, C., A. Nemirovskii, and Yu. Nesterov (1995). New Variants of Bundle Methods. *Mathematical Programming* **69**, 111-147.

[25] Linderoth, J.T. and S.J. Wright (2003). Decomposition Algorithms for Stochastic Programming on a Computational Grid *Computational Optimization and Applications* **24**, 207-250.

[26] Luedtke, J. (2008). New formulations for optimization under stochastic dominance constraints. *SIAM Journal on Optimization* **19**, 1433-1450.

[27] Meskarian, R., H. Xu, and J. Fliege (2012). Numerical methods for stochastic programs with second order dominance constraints with applications to portfolio optimization. *European Journal of Operational Research* **216**, 376-385.

[28] Mitra, G. M. Tamiz and J. Yadegar (1988). Experimental investigation of an interior search method within a simplex framework. *Communications of the ACM* **31**, 1474-1482.

[29] Murty, K.G. and Y. Fathi (1984). A feasible direction method for linear programming. *Operations Research Letters* **3**, 121-127.

[30] Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers and Operations Research* **39**, 541-559.

[31] Ogryczak, W. (2000). Multiple criteria linear programming model for portfolio selection. *Annals of Operations Research* **97**, 143-162.

[32] Ogryczak, W. and A. Ruszczyński (2002). Dual stochastic dominance and related mean-risk models. *SIAM Journal on Optimization* **13**, 60-78.

[33] Oliveira, W. and C. Sagastizábal (2012). Level bundle methods for oracles with on-demand accuracy. *Optimization Online,* March 2012.

[34] Oliveira, W. C. Sagastizábal, and S. Scheimberg. (2011). Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization* **21**, 517-544.

[35] Pflug, G. (2000). Some remarks on the value-at-risk and the conditional value-at-risk. In *Probabilistic Constrained Optimization: Methodology and Applications.* S. Uryasev, ed. Kluwer Academic Publishers, Norwell, MA.

[36] Rockafellar, R.T. and S. Uryasev (2000). Optimization of conditional value-at-risk. *Journal of Risk* **2**, 21-41.

[37] Rockafellar, R.T. and S. Uryasev (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance* **26**, 1443-1471.

[38] Roman, D., K. Darby-Dowman, and G. Mitra (2006). Portfolio construction based on stochastic dominance and target return distributions. *Mathematical Programming* **108**, 541-569.

[39] Rudolf, G. and A. Ruszczyński (2008). Optimization problems with second order stochastic dominance constraints: duality, compact formulations, and cut generation methods. *SIAM Journal on Optimization* **19**, 1326-1343.

[40] Ruszczyński, A. (1986). A Regularized Decomposition Method for Minimizing the Sum of Polyhedral Functions. *Mathematical Programming* **35**, 309-333.

[41] Trukhanov S., L. Ntaimo, and A. Schaefer (2010). Adaptive multicut aggregation for two-stage stochastic linear programs. *European Journal of Operational Research* **206**, 395-406.

[42] Van Slyke, R. and R.J.-B. Wets (1969). L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics* **17**, 638-663.

[43] Vespucci, M.T., M. Bertocchi, L. Escudero, M. Innorta, and S. Zigrino. A stochastic model for generation expansion planning in the long period with different risk measures. In: *Applied Mathematical Optimization and Modelling. Extended Abstracts of the APMOD 2012 Conference* (L. Suhl, G. Mitra, C. Lucas, A. Koberstein, and L. Beckmann, editors), 114-121. Vol 8 of the DSOR Contributions to Information Sytems. DS&OR Lab, University of Paderborn.

[44] Whitmore, G.A. and M.C. Findlay (1978). *Stochastic Dominance: An Approach to Decision-Making Under Risk.* D.C.Heath, Lexington, MA.

[45] Wolf, C. and A. Koberstein (2012). Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method. *Optimization Online,* September 2012.

[46] Zverovich, V. C.I. Fábián, E.F.D. Ellison, and G. Mitra. A computational study of a solver system for processing two-stage stochastic LPs with enhanced Benders decomposition. *Mathematical Programming Computation* **4** (2012), 211-238.