A randomized method for handling a difficult function in a convex optimization problem, motivated by probabilistic programming

Csaba I. Fábián* Tamás Szántai†

Abstract

We propose a randomized gradient method for the handling of a convex function whose gradient computation is demanding. The method bears a resemblance to the stochastic approximation family. But in contrast to stochastic approximation, the present method builds a model problem.

The approach requires that estimates of function values and gradients be provided at the iterates. We present a variance reduction Monte Carlo simulation procedure to provide such estimates in the case of certain probabilistic functions.

Keywords. Convex optimization, stochastic optimization, probabilistic problems.

1 Introduction

We deal with approximate methods for the solution of smooth convex programming problems. First we consider minimization over a polyhedron:

$$\min\ \phi(T\boldsymbol{x}) \quad \text{subject to} \quad A\boldsymbol{x} \leq \boldsymbol{b}, \tag{1}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a convex function whose gradient computation is demanding. The vectors are $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{b} \in \mathbb{R}^r$, and the matrices T and A are of sizes $n \times m$ and $r \times m$, respectively. For the sake of simplicity we assume that the feasible domain is not empty and is bounded. We then consider the minimization of a linear cost function subject to a difficult convex constraint:

$$\min\ \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to} \quad \breve{A}\boldsymbol{x} \leq \breve{\boldsymbol{b}}, \quad \phi(T\boldsymbol{x}) \leq \pi, \tag{2}$$

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

The motivation for the above forms are the classic probability maximization and probabilistic constrained problems, where $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ with a logconcave distribution function $F(\boldsymbol{z})$. In [7], an inner approximation was proposed for the probabilistic function. The approach proved easy to implement and invulnerable to noise in gradient computation. – Noisy gradient estimates may yield iterates that do not improve much on our current model. But we retain a true inner approximation, provided objective values at new iterates are evaluated with appropriate accuracy.

Let us briefly overview a couple of closely related probabilistic programming approaches. – For a broader survey, see [7] and references therein. – Given a distribution and a number p $(0 < p < 1)$, a probabilistic constraint confines search to the level set $\mathcal{L}(F, p) = \{\, \boldsymbol{z} \,|\, F(\boldsymbol{z}) \geq p \,\}$ of the distribution function $F(\boldsymbol{z})$. Prékopa

*Department of Informatics, Faculty of Engineering and Computer Science, Kecskemét College, Pallasz Athéné University. Izsáki út 10, 6000 Kecskemét, Hungary. Email: fabian.csaba@gamf.kefo.hu.

†Department of Differential Equations, Faculty of Natural Sciences, Budapest University of Technology and Economics. Műegyetem rkp 3-9, 1111 Budapest, Hungary. Email: szantai@math.bme.hu.

[18] initiated a novel solution approach by introducing the concept of p-efficient points. $\boldsymbol{z}$ is p-efficient if $F(\boldsymbol{z}) \geq p$ and there exists no $\boldsymbol{z}'$ such that $\boldsymbol{z}' \leq \boldsymbol{z}$, $\boldsymbol{z}' \neq \boldsymbol{z}$, $F(\boldsymbol{z}') \geq p$. Prékopa, Vizvári, and Badics [20] consider problems with random parameters having a discrete finite distribution. They first enumerate p-efficient points, and based on these, build a convex relaxation of the problem.

Dentcheva, Prékopa, and Ruszczyński [6] formulate the probabilistic constraint in a split form: $T\boldsymbol{x} = \boldsymbol{z}$, where $\boldsymbol{z}$ belongs to the level set $\mathcal{L}(F, p)$; and construct a Lagrangian dual by relaxing the constraint $T\boldsymbol{x} = \boldsymbol{z}$. The dual functional is the sum of two functionals that are respective optimal objective value functions of two simpler problems. The first auxiliary problem is a linear programming problem, and the second one is the minimization of a linear function over the level set $\mathcal{L}(F, p)$. Based on this decomposition, the authors develop a method, called cone generation, that finds new p-efficient points in course of the optimization process.

[7] focusses on probability maximization. A polyhedral approximation is constructed to the epigraph of the probabilistic function. This is analogous to the use of p-efficient points, and the approach was motivated by that concept. The dual problem is constructed and decomposed in the manner of [6], but the nonlinear subproblem is easier. In [6], finding a new p-efficient point amounts to minimization over the level set $\mathcal{L}(F, p)$. In contrast, a new approximation point in [7] is found by unconstrained minimization. Moreover, a practical approximation scheme was developed in [7]: instead of exactly solving an unconstrained subproblem occurring during the process, just a single line search was made in each case. Implementation based on this approximation scheme proved quite robust, and a theoretic explanation for this behavior was also found.

In the present paper, we extend the approach of [7] to handling gradient estimates. We also propose simulation schemes to obtain such estimates in case of probabilistic problems. We present the inner approximation approach in an idealized setting:

Assumption 1 *The function $\phi(\boldsymbol{z})$ is twice continuously differentiable, and the Hessian matrix satisfies*

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

with some $\alpha, \omega \in \mathbb{R}$ $(0 < \alpha \leq \omega)$. Here I is the identity matrix, and the relation $U \preceq V$ between matrices means that $V - U$ is positive semidefinite.

In Sections 2 and 3 we present a brisk overview of the models and the column generation approach of [7]. In Section 4 we present the column-generation approach from a dual viewpoint, as a cutting-plane method. (The dual viewpoint has the advantage that the cutting-plane method can be regularized, but we do not consider regularization in this paper.)

The cutting-plane model of the dual approach – like the inner approximation of the primal one – is invulnerable to gradient computation errors. This feature facilitates the use of gradient estimates. In Section 5 we extend the method in this direction. The motivation for applying gradient estimates was our computational experience reported in [7]: most of the computation effort was spent in computing gradients. – In that computational study we solved classic probability maximization problems; namely, we had $\phi(\boldsymbol{z}) = -\log F(\boldsymbol{z})$ with a multivariate normal distribution function $F(\boldsymbol{z})$. Given an n-dimensional normal distribution, a component of the gradient $\nabla F(\boldsymbol{z})$ was obtained as the product of an appropriate $(n-1)$-dimensional normal distribution function value and a univariate normal density function value (see the formula in Section 6.6.4 of Prékopa's book [19]). The numerical computation of multivariate normal distribution function values was performed by Genz's subroutine [11]. In our study, most of the computation time was spent in the Genz subroutine. Most demanding were the gradient computations, each requiring n calls to the Genz subroutine. – We conclude that easily computable estimates for the gradients are well worth using, even if the iteration count increases due to estimation errors.

For the estimation of function values and gradients in case of a probabilistic objective, we present a variance reduction Monte Carlo simulation procedure in Section 6. This procedure is applicable to gradient estimation in case of normal, Dirichlet, and t-distributions.

In Section 7 we deal with the convex constrained problem (2). Well-known approximation schemes for this problem consist of the solution of a sequence of problems of the form (1). We are going to show that the approximation tools described in previous sections facilitate such a solution scheme in the present case.

2 Problem and model formulation

In this section we formulate the dual problem and construct polyhedral models of the primal and dual problems. We follow the construction in [7], details can be found in that paper. Though in [7], we exploited monotonicity of the probabilistic objective and variable splitting was based on $\boldsymbol{z} \leq T\boldsymbol{x}$. In the present paper, we apply the traditional form of variable splitting: Problem (1) will be written as

$$\min\ \phi(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \ \boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}. \tag{3}$$

This problem has an optimal solution due to our assumption that on feasible domain of (1). Introducing the multiplier vector $-\boldsymbol{y} \in \mathbb{R}^r, -\boldsymbol{y} \geq \boldsymbol{0}$ to the constraint $A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0}$, and $-\boldsymbol{u} \in \mathbb{R}^n$ to the constraint $\boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}$, the Lagrangian dual of (3) can be written as

$$\max\ \{\boldsymbol{y}^T\boldsymbol{b} - \phi^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}, \tag{4}$$

where

$$\mathcal{D} := \ \left\{\ (\boldsymbol{y}, \boldsymbol{u}) \in \mathbb{R}^{r+n}\ \mid\ \boldsymbol{y} \leq \boldsymbol{0},\ \ T^T\boldsymbol{u} = A^T\boldsymbol{y}\ \right\}. \tag{5}$$

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

2.1 Polyhedral models

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

$$\phi_k(\boldsymbol{z}) = \min \sum_{i=0}^{k} \lambda_i \phi_i \quad \text{such that} \quad \lambda_i \geq 0\ (i = 0, \ldots, k),\ \ \sum_{i=0}^{k} \lambda_i = 1, \quad \sum_{i=0}^{k} \lambda_i \boldsymbol{z}_i = \boldsymbol{z}. \tag{6}$$

If $\boldsymbol{z} \notin \mathrm{Conv}(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k)$, then let $\phi_k(\boldsymbol{z}) := +\infty$. A polyhedral model of Problem (3) is

$$\min\ \phi_k(\boldsymbol{z}) \quad \text{subject to} \quad A\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0},\ \boldsymbol{z} - T\boldsymbol{x} = \boldsymbol{0}. \tag{7}$$

We assume that (7) is feasible, i.e., its optimum is finite. This can be ensured by proper selection of the initial $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k$ points. The convex conjugate of $\phi_k(\boldsymbol{z})$ is

$$\phi_k^\star(\boldsymbol{u}) = \max_{0 \leq i \leq k}\ \{\boldsymbol{u}^T\boldsymbol{z}_i - \phi_i\}. \tag{8}$$

As $\phi_k^\star(.)$ is a cutting-plane model of $\phi^\star(.)$, the following problem is a polyhedral model of Problem (4):

$$\max\ \{\boldsymbol{y}^T\boldsymbol{b} - \phi_k^\star(\boldsymbol{u})\} \quad \text{subject to} \quad (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D}. \tag{9}$$

2.2 Linear programming formulations

The primal model problem (6)-(7) will be formulated as

$$\begin{array}{lll}
\min & \sum\limits_{i=0}^{k} \phi_i \lambda_i & \\
\text{such that} & \lambda_i \geq 0 & (i = 0, \ldots, k), \\
& \sum\limits_{i=0}^{k} \lambda_i & = 1, \\
& \sum\limits_{i=0}^{k} \lambda_i \boldsymbol{z}_i \quad -T\boldsymbol{x} & = \boldsymbol{0}, \\
& A\boldsymbol{x} & \leq \boldsymbol{b}.
\end{array} \tag{10}$$

The dual model problem (8)-(9), formulated as a linear programming problem, is just the LP dual of (10):

$$\begin{array}{lllll} \max & \vartheta & + & \boldsymbol{b}^T\boldsymbol{y} & \\ \text{such that} & & & \boldsymbol{y} & \leq \mathbf{0}, \\ & \vartheta & + \boldsymbol{z}_i^T\boldsymbol{u} & & \leq \phi_i \qquad (i=0,\ldots,k), \\ & & -T^T\boldsymbol{u} & + A^T\boldsymbol{y} & = \mathbf{0}. \end{array} \tag{11}$$

Let $(\overline{\lambda}_0,\ldots,\overline{\lambda}_k,\overline{\boldsymbol{x}})$ and $(\overline{\vartheta},\overline{\boldsymbol{u}},\overline{\boldsymbol{y}})$ denote respective optimal solutions of the problems (10) and (11) – both existing due to our assumption concerning the feasibility of (7) and hence (10). Let moreover

$$\overline{\boldsymbol{z}} = \sum_{i=0}^{k} \overline{\lambda}_i \boldsymbol{z}_i. \tag{12}$$

Observation 2 *We have* $\phi_k(\overline{\boldsymbol{z}}) = \sum_{i=0}^{k} \phi_i\overline{\lambda}_i = \overline{\vartheta} + \overline{\boldsymbol{u}}^T\overline{\boldsymbol{z}}$.

The first equality follows from the equivalence of (10) on the one hand, and (6)-(7) on the other hand. The second equality is a straight consequence of complementarity.

Observation 3 *We have* $\overline{\vartheta} = -\phi_k^\star(\overline{\boldsymbol{u}})$.

This follows from the equivalence between (11) on the one hand and (8)-(9) on the other hand.

Remark 4 *A consequence of Observations 2 and 3 is* $\phi_k(\overline{\boldsymbol{z}}) + \phi_k^\star(\overline{\boldsymbol{u}}) = \overline{\boldsymbol{u}}^T\overline{\boldsymbol{z}}$. *This is Fenchel's equality between* $\overline{\boldsymbol{u}}$ *and* $\overline{\boldsymbol{z}}$, *with respect to the model function* $\phi_k(.)$.

3 Primal viewpoint: column generation

In [7], the probability maximization problem is solved by iteratively adding improving columns to the primal model. In this section we give a brisk overview of the practical approximation scheme proposed in that paper.

An optimal dual solution (i.e., shadow price vector) of the current model problem is $(\overline{\vartheta},\overline{\boldsymbol{u}},\overline{\boldsymbol{y}})$. Given a vector $\boldsymbol{z}\in\mathbb{R}^n$, we can add a new column in (10), corresponding to $\boldsymbol{z}_{k+1}=\boldsymbol{z}$. This is an improving column if its reduced cost

$$\overline{\rho}(\boldsymbol{z}) := \overline{\vartheta} + \overline{\boldsymbol{u}}^T\boldsymbol{z} - \phi(\boldsymbol{z}) \tag{13}$$

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

$$\overline{\rho}(\overline{\boldsymbol{z}}) \geq \overline{\vartheta} + \overline{\boldsymbol{u}}^T\overline{\boldsymbol{z}} - \phi_k(\overline{\boldsymbol{z}}) = 0 \tag{14}$$

follows from $\phi_k(.)\geq\phi(.)$ and Observation 2.

In the context of the simplex method, the Markowitz rule is a well-known and often-used rule of column selection. The Markowitz rule selects the vector with the largest reduced cost. Coming back to the present problem (10), let

$$\overline{\mathcal{R}} := \max_{\boldsymbol{z}} \overline{\rho}(\boldsymbol{z}). \tag{15}$$

The column with the largest reduced cost can, in theory, be found by a steepest descent method applied to the function $-\overline{\rho}(\boldsymbol{z})$. Though finding a near-optimal solution proved rather time-consuming in the computational study of [7]. As a practical alternative, only a single line search was performed, starting from $\overline{\boldsymbol{z}}$. This simple method proved effective and robust. Moreover, a theoretical explanation was also found for the efficiency of the approach, based on the following well-known theorem:

Theorem 5 *Let Assumption 1 hold for the function $f : \mathbb{R}^n \to \mathbb{R}$. Let us minimize $f(\boldsymbol{z})$ over $\mathbb{R}^n$ using a steepest descent method, starting from a point $\boldsymbol{z}^0$. Let $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^j, \ldots$ denote the iterates obtained by applying exact line search at each step. Then we have*

$$f\left(\boldsymbol{z}^j\right) - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega}\right)^j \left[\, f\left(\boldsymbol{z}^0\right) - \mathcal{F} \,\right], \tag{16}$$

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

Proof of this theorem can be found in, e.g., [22], [15]. The following corollary was obtained in [7]:

Corollary 6 *Let β $(0 < \beta \ll 1)$ be given. A finite (and moderate) number of steps with the steepest descent method results a vector $\widehat{\boldsymbol{z}}$ such that*

$$\overline{\rho}\,(\widehat{\boldsymbol{z}}) \;\geq\; (1 - \beta)\,\overline{\mathcal{R}}. \tag{17}$$

This can be proven by substituting $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z})$, $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ in (16), and applying (14). Selecting j such that $\left(1 - \frac{\alpha}{\omega}\right)^j \leq \beta$ yields an appropriate $\widehat{\boldsymbol{z}} = \boldsymbol{z}^j$.

In view of the Markowitz rule mentioned above, the vector $\widehat{\boldsymbol{z}}$ in Corollary 6 is a fairly good improving vector in the column generation scheme.

In the computational study of [7], just a single line search was performed in each reduced cost maximization; i.e., $j = 1$ according to the notation of Theorem 5. (Even this single line search was inexact, making a limited number of steps in the direction of steepest ascent.) Our implementation proved reliable even with this simple procedure.

In case of probabilistic functions, Assumption 1 does not hold for every $\boldsymbol{z} \in \mathbb{R}^n$. Our computational experience in [7] was, however, that the probabilistic objectives were well conditioned over certain domains. The iterates obtained by the above approximation procedure always remained in the respective safe domains.

Remark 7 *To check near-optimality of the current solution, we can use the usual LP stopping rule: $\overline{\mathcal{R}}$ should be less than a fixed optimality tolerance. – For the present special linear programming problem (10), this is not just a heuristic rule; as we show in the next section, $\overline{\mathcal{R}}$ is actually a bound on the gap between the respective optima of the model problem and the original convex problem.*

If we have good estimates for α and ω in Assumption 1, then $\overline{\mathcal{R}}$ can be estimated on the basis of Corollary 6. Hence the gap can be kept under effective control in course of the solution process.

If no reliable estimates for α and ω are known, then Corollary 6 is just a theoretical justification for limiting the numbers of the line searches in each steepest ascent method. The column generation procedure is terminated if $\|\nabla\overline{\rho}(\overline{\boldsymbol{z}})\|$ is small. We can construct an upper bound of the final gap $\overline{\mathcal{R}}$ using the gradient.

4 Dual viewpoint: cutting planes

The simplex method can be viewed as a cutting-plane method. This fact has been part of the professional folklore ever since the simplex method became widely known. Simplex and cutting-plane methods are parallelly discussed in Section 3.4 of Prékopa's book [19]. A closer description of the present situation can be found in [9], Section 4.

4.1 Dimension reduction

For the sake of simplicity, let us make

Assumption 8 *The inequality system $A\boldsymbol{x} \leq \boldsymbol{b}$ contains box constraints in the form of $\underline{\boldsymbol{b}} \leq \boldsymbol{x} \leq \overline{\boldsymbol{b}}$, where $\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}} \in \mathbb{R}^r$ are given vectors $(\underline{\boldsymbol{b}} \leq \overline{\boldsymbol{b}})$.*

I.e., we have

$$A = \begin{pmatrix} A' \\ I \\ -I \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}' \\ \overline{\boldsymbol{b}} \\ -\underline{\boldsymbol{b}} \end{pmatrix}. \tag{18}$$

Let

$$\mu(\boldsymbol{u}) \;\; = \min \{\, -\boldsymbol{b}^T\boldsymbol{y} \mid (\boldsymbol{y}, \boldsymbol{u}) \in \mathcal{D} \,\}. \tag{19}$$

This function is defined for every $\boldsymbol{u}$ since $T^T\boldsymbol{u} = A^T\boldsymbol{y}$ is solvable in $\boldsymbol{y}$ ($\boldsymbol{y} \leq \boldsymbol{0}$) due to A having the form (18). We will formulate the dual problems using the function $\mu(\boldsymbol{u})$. For convenience, we transform the problems into minimization forms. The original dual problem (4) assumes the form

$$\min \{\phi^\star(\boldsymbol{u}) + \mu(\boldsymbol{u})\}, \tag{20}$$

and the model problem (9) assumes the form

$$\min \{\phi_k^\star(\boldsymbol{u}) + \mu(\boldsymbol{u})\}. \tag{21}$$

4.2 Cut generation

Let $\overline{\boldsymbol{u}}$ be a minimizer of (21), in accordance with our notation in former sections. We are going to compute an approximate support function to $\phi^\star(\boldsymbol{u})$ at $\overline{\boldsymbol{u}}$. This will be of the form

$$\ell(\boldsymbol{u}) := \boldsymbol{u}^T\boldsymbol{z} - \phi\,(\boldsymbol{z})\,, \tag{22}$$

with an appropriate vector $\boldsymbol{z}$. We have $\ell(\boldsymbol{u}) \leq \phi^\star(\boldsymbol{u})$ for any $\boldsymbol{u}$ by the definition of $\phi^\star(\boldsymbol{u})$. We are going to compute $\boldsymbol{z}$ such that $\phi^\star(\overline{\boldsymbol{u}}) - \ell(\overline{\boldsymbol{u}})$ be relatively small.

Using support functions of the above form, a cutting-plane scheme for the problem (20) is easily implemented. We build a polyhedral model $\phi_k^\star(\boldsymbol{u})$ of $\phi^\star(\boldsymbol{u})$ – always adding the appropriate $\boldsymbol{z}$ vector to the dual model as $\boldsymbol{z}_{k+1}$. On the other hand, we will work with $\mu(\boldsymbol{u})$ as a polyhedral model of itself. This is a workable setup; the current model function can be minimized by just solving the linear programming problem (11) – take into account Observation 3.

Coming back to the construction of the approximate support function (22), we wish to construct $\ell(\boldsymbol{u})$ whose graph cuts deeply into the epigraph of the model function $\phi_k^\star(\boldsymbol{u})$. (Depth being measured at $\overline{\boldsymbol{u}}$.) I.e., we wish the following difference to be large:

$$\ell(\overline{\boldsymbol{u}}) - \phi_k^\star(\overline{\boldsymbol{u}}) \;=\; \overline{\boldsymbol{u}}^T\boldsymbol{z} - \phi(\boldsymbol{z}) \;- \phi_k^\star(\overline{\boldsymbol{u}}) \;=\; \overline{\rho}(\boldsymbol{z}), \tag{23}$$

where the second equality follows from Observation 3. (This is a dual interpretation the reduced cost.)

Let $\Psi^\star$ denote the minimum of (20). Let us further introduce the notation

$$\underline{\Psi} := \phi_k^\star(\overline{\boldsymbol{u}}) + \mu(\overline{\boldsymbol{u}}) \qquad \text{and} \qquad \overline{\Psi} := \phi^\star(\overline{\boldsymbol{u}}) + \mu(\overline{\boldsymbol{u}}). \tag{24}$$

Obviously we have $\underline{\Psi} \;\leq\; \Psi^\star \;\leq\; \overline{\Psi}$. As for the gap between the upper and the lower bound, we have

$$\overline{\Psi} - \underline{\Psi} \;=\; \phi^\star(\overline{\boldsymbol{u}}) - \phi_k^\star(\overline{\boldsymbol{u}}) \;=\; \max_{\boldsymbol{z}} \overline{\rho}(\boldsymbol{z}) \;=\; \overline{\mathcal{R}}, \tag{25}$$

where the second equality follows from the definition of the conjugate function, and the third equality is in accordance with our former notation.

In order to construct inexact cuts, let us consider a dual form of Corollary 6:

Proposition 9 *Let β $(0 < \beta \ll 1)$ be given. We can construct a linear function $\hat{\ell}(\boldsymbol{u})$ such that*

$$\hat{\ell}(\boldsymbol{u}) \leq \phi^\star(\boldsymbol{u}) \quad \textit{holds for any } \boldsymbol{u}, \;\; \textit{and}$$

$$\hat{\ell}\,(\overline{\boldsymbol{u}}) \geq \phi^\star\,(\overline{\boldsymbol{u}}) - \beta\overline{\mathcal{R}}.$$

In words: $\hat{\ell}(\boldsymbol{u})$ is an approximate support function to $\phi^\star(\boldsymbol{u})$ at $\overline{\boldsymbol{u}}$. The difference between the function values at the current iterate is bounded by the portion $\beta\,\overline{\mathcal{R}}$ of the gap.

Proof. According to Corollary 6, we can construct a vector $\widehat{\boldsymbol{z}}$ such that (17) holds. Using this $\widehat{\boldsymbol{z}}$, let us define $\hat{\ell}(\boldsymbol{u})$ according to (22). Then we have

$$\begin{aligned} \hat{\ell}\,(\overline{\boldsymbol{u}}) &= \overline{\rho}\,(\widehat{\boldsymbol{z}}) + \phi_k^\star\,(\overline{\boldsymbol{u}}) && \text{due to (23)} \\ &\geq (1-\beta)\,\overline{\mathcal{R}} + \phi_k^\star\,(\overline{\boldsymbol{u}}) && \text{due to (17)} \\ &= \phi^\star\,(\overline{\boldsymbol{u}}) - \beta\,\overline{\mathcal{R}} && \text{due to (25).} \end{aligned} \tag{26}$$

5 Working with gradient estimates

In this section we show that the column generation scheme of [7] (sketched in Section 3), and the cutting-plane scheme of Section 4 can be implemented as a randomized method using gradient estimates.

We wish to minimize $-\overline{\rho}(\boldsymbol{z})$ over $\mathbb{R}^n$. Given $\boldsymbol{z}^\circ \in \mathbb{R}^n$, let $\boldsymbol{g}^\circ = -\nabla\overline{\rho}(\boldsymbol{z}^\circ)$.

Assumption 10 *Let $\sigma > 0$ be given. We can construct a realization of a random vector $\boldsymbol{G}^\circ$ satisfying*

$$E(\boldsymbol{G}^\circ) = \boldsymbol{g}^\circ \qquad \textit{and} \qquad E\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) \;\leq\; \sigma\,\|\boldsymbol{g}^\circ\|^2\,. \tag{27}$$

From (27) follows

$$\mathrm{E}\left(\|\boldsymbol{G}^\circ\|^2\right) \;=\; \mathrm{E}\left(\|\boldsymbol{G}^\circ - \boldsymbol{g}^\circ\|^2\right) + \|\boldsymbol{g}^\circ\|^2 \;\leq\; (\sigma+1)\,\|\boldsymbol{g}^\circ\|^2\,. \tag{28}$$

Let us consider the following randomized form of Theorem 5:

Theorem 11 *Let Assumption 1 hold for the function $f : \mathbb{R}^n \to \mathbb{R}$. Let us minimize $f(\boldsymbol{z})$ over $\mathbb{R}^n$. We perform a steepest descent method using gradient estimates. (Given an iterate $\boldsymbol{z}^\circ$, a gradient estimate $\boldsymbol{G}^\circ$ is generated and a line search is performed in that direction.) We assume that gradient estimates at the respective iterates are generated independently, and (27) - (28) hold for each of them.*

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

$$E\left[f\left(\boldsymbol{z}^j\right)\right] - \mathcal{F} \;\leq\; \left(1 - \frac{\alpha}{\omega(\sigma+1)}\right)^j \left(f\left(\boldsymbol{z}^0\right) - \mathcal{F}\right), \tag{29}$$

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

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

To begin with, let us focus on the first line search whose starting point is $\boldsymbol{z}^\circ = \boldsymbol{z}^0$. Here $\boldsymbol{z}^\circ$ is a given (not random) vector. We are going to adopt the usual proof of Theorem 5 to employing the gradient estimate $\boldsymbol{G}^\circ$ instead of the gradient $\boldsymbol{g}^\circ$. From $\nabla^2 f(\boldsymbol{z}) \preceq \omega I$, it follows that

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

holds for any $t \in \mathbb{R}$. Considering expectations in both sides, we get

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

according to (28). Let us consider the respective minima in t separately of the two sides. The right-hand side is a quadratic expression, yielding minimum at $t = \frac{1}{\omega(\sigma+1)}$. Inequality is inherited to minima, hence

$$\min_t \mathrm{E}\left[f\left(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ\right)\right] \;\leq\; f\left(\boldsymbol{z}^\circ\right) \;-\; \frac{1}{2\omega(\sigma+1)}\,\|\boldsymbol{g}^\circ\|^2\,. \tag{30}$$

For the left-hand side, we obviously have

$$\mathrm{E}\left[\min_t f(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ)\right] \;\le\; \min_t \mathrm{E}\left[f(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ)\right]. \tag{31}$$

(This is analogous to the basic inequality comparing the wait-and-see and the here-and-now approaches for classic two-stage stochastic programing problems.)

Let $\boldsymbol{z}'$ denote the minimizer of the line search in the left-hand side of (31), i.e., $f(\boldsymbol{z}') = \min_t f(\boldsymbol{z}^\circ - t\boldsymbol{G}^\circ)$. (Of course $\boldsymbol{z}'$ is a random vector since it depends on $\boldsymbol{G}^\circ$.) Substituting this in (31) and comparing with (30), we get

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

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

$$\mathrm{E}\left[f(\boldsymbol{z}')\right] - \mathcal{F} \;\le\; f(\boldsymbol{z}^\circ) - \mathcal{F} \;-\; \frac{1}{2\omega(\sigma+1)}\,\|\boldsymbol{g}^\circ\|^2. \tag{32}$$

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

$$\|\boldsymbol{g}^\circ\|^2 \;\ge\; 2\alpha\,(f(\boldsymbol{z}^\circ) \;-\mathcal{F})$$

(see the classic proof of Theorem 5). Combining this with (32), we get

$$\mathrm{E}\left[f(\boldsymbol{z}')\right] - \mathcal{F} \;\le\; f(\boldsymbol{z}^\circ) - \mathcal{F} \;-\; \frac{\alpha}{\omega(\sigma+1)}\,(f(\boldsymbol{z}^\circ) \;-\mathcal{F}) \;=\; \left(1 - \frac{\alpha}{\omega(\sigma+1)}\right)(f(\boldsymbol{z}^\circ) \;-\mathcal{F}). \tag{33}$$

As we have assumed that $\boldsymbol{z}^\circ$ is a given (not random) vector, the right-hand side of (33) is deterministic, and the expectation in the left-hand side is considered according to the distribution of $\boldsymbol{G}^\circ$.

Let us now examine the $(l+1)$th line search (for $1 \le l \le j-1$) where the starting point is $\boldsymbol{z}^\circ = \boldsymbol{z}^l$ and the minimizer is $\boldsymbol{z}' = \boldsymbol{z}^{l+1}$. Of course (33) holds with these objects also, but now both sides are random variables, depending on the vectors $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^{l-1}$. (The expectation in the left-hand side is a conditional expectation.) Let us consider the respective expectations of the two sides, according to the joint distribution of $\boldsymbol{G}^0, \ldots, \boldsymbol{G}^{l-1}$. As the random gradient vectors were generated independently, we get

$$\mathrm{E}\left[f\left(\boldsymbol{z}^{l+1}\right)\right] - \mathcal{F} \;\le\; \left(1 - \frac{\alpha}{\omega(\sigma+1)}\right)\left(\mathrm{E}\left[f\left(\boldsymbol{z}^{l}\right)\right] \;-\mathcal{F}\right), \tag{34}$$

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

Finally, (29) follows from the iterative application of (34). □

Corollary 12 *Let a tolerance β $(0 < \beta \ll 1)$ and a probability p $(0 < p \ll 1)$ be given. A finite (and moderate) number of steps with the above randomized steepest descent method results a vector $\widehat{\boldsymbol{z}}$ such that*

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

I.e., with a high probability, $\widehat{\boldsymbol{z}}$ is a fairly good improving vector in the column generation scheme.

Proof. Substituting $f(\boldsymbol{z}) = -\overline{\rho}(\boldsymbol{z})$ and $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ in (29) and taking into account (14), we get

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

with $\varrho = 1 - \frac{\alpha}{\omega(\sigma+1)}$. The gap $\overline{\mathcal{R}}$ is obviously non-negative. In case $\overline{\mathcal{R}} = 0$, the starting iterate $\boldsymbol{z}^0 = \overline{\boldsymbol{z}}$ of the steepest descent method was already optimal, due to (14). In what follows we assume $\overline{\mathcal{R}} > 0$. A trivial transformation results

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

By Markov's inequality, we get

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

and a trivial transformation yields

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

Hence

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

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

Gradients of the function $-\overline{\rho}(\boldsymbol{z})$ have the form $\nabla\phi(\boldsymbol{z}) - \overline{\boldsymbol{u}}$. As the procedure progresses, the difference $\nabla\phi(\widehat{\boldsymbol{z}}) - \overline{\boldsymbol{u}}$ gets small. To satisfy the requirement (27) on variance, better and better estimates are needed. We can control accuracy like in the deterministic column generation method of Section 3.

Remark 13 *If we have good estimates for α and ω in Assumption 1, then $\overline{\mathcal{R}}$ can be estimated on the basis of Corollary 12. Otherwise Corollary 12 is just a theoretical justification for limiting the numbers of the line searches in each steepest ascent procedure.*

When the column generation scheme stops, we need to check the magnitude of the current gap. It may happen that only a statistical verification is possible, on the basis of Assumption 10.

A dual form of the above corollary is

Proposition 14 *Let a tolerance β $(0 < \beta \ll 1)$ and a probability p $(0 < p \ll 1)$ be given. We can construct a random linear function $\hat{\ell}(\boldsymbol{u})$ such that*

$$\hat{\ell}(\boldsymbol{u}) \leq \phi^\star(\boldsymbol{u}) \quad \textit{holds for any } \boldsymbol{u}, \;\; \textit{and}$$

$$P\Big(\, \hat{\ell}\,(\overline{\boldsymbol{u}}) \;\geq\; \phi^\star\,(\overline{\boldsymbol{u}}) - \beta\,\overline{\mathcal{R}} \,\Big) \;\geq\; 1 - p.$$

The proof is the same as that of Proposition 9, but use Corollary 12 instead of Corollary 6.

6 Easily computable estimates of function values and gradients

For the partial derivatives of any multivariate probability distribution function we have the general formula

$$\frac{\partial F(z_1,\ldots,z_n)}{\partial z_i} = F(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n|\; z_i) f_i(z_i) \tag{35}$$

where $F(z_1,\ldots,z_n)$ is the probability distribution function of the random variables $\xi_1,\ldots,\xi_n$, $F(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n|\; z_i)$ is the conditional probability distribution function of the random variables $\xi_1,\ldots,\xi_{i-1},\xi_{i+1},\ldots,\xi_n$, given that $\xi_i = z_i$ and $f_i(z)$ is the probability density function of the random variable ξ_i (see Formula (6.6.22) on the page 203 of Prékopa's book [19]). From Formula (35) it follows that if the multivariate probability distribution at issue has conditional probability distributions of its own type then we can calculate the multivariate probability distribution function values and its partial derivatives by the same procedure. Such type of multivariate probability distributions are for example the multivariate normal, the multivariate t-distribution and the Dirichlet distribution.

Therefore in this section we present a variance reduction Monte Carlo simulation procedure for the estimation of multivariate probability distribution function values, only. The procedure was proposed in Szántai's thesis [23], and quoted in Sections 6.5 and 6.6 of Prékopa's book [19].

This procedure can be applied to any multivariate probability distribution function. The only condition is that we have to be able to calculate the one- and the two-dimensional marginal probability distribution function values. Accuracy can easily be controlled by changing the sample size.

This way we can construct gradient estimates satisfying Assumption 10.

As we have

$$F(z_1,\dots,z_n) = \mathrm{P}\left(\xi_1 < z_1,\dots,\xi_n < z_n\right) = 1 - \mathrm{P}(\overline{A}_1 \cup \dots \cup \overline{A}_n),$$

where

$$\overline{A}_i = \{\xi_i \geq z_i\} \quad (i = 1,\dots,n),$$

we can apply bounding and simulation results for the probability of union of events.

6.1 Crude Monte Carlo simulation

If μ denotes the number of those events which occur out of the events $\overline{A}_1, \overline{A}_2, \dots, \overline{A}_n$, then the random variable

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

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

6.2 Monte Carlo simulation of the differences between the true probability value and its respective second order Boole–Bonferroni bounds

For the probability $\overline{P} = \mathrm{P}(\overline{A}_1 \cup \overline{A}_2 \cup \dots \cup \overline{A}_n)$ we have the so called second order Boole–Bonferroni bounds (see Prékopa's book [19]).

The lower bound is

$$L_2 = \frac{2}{k^*+1}\overline{S}_1 - \frac{2}{k^*(k^*+1)}\overline{S}_2,$$

where

$$k^* = 1 + \left\lfloor \frac{2\overline{S}_2}{\overline{S}_1} \right\rfloor.$$

The upper bound is

$$U_2 = \overline{S}_1 - \frac{2}{n}\overline{S}_2.$$

In both cases $\overline{S}_1$ and $\overline{S}_2$ are the first resp. second order binomial moments of the random variable μ which can be expressed also as

$$\overline{S}_1 = \sum_{i=1}^{n} \mathrm{P}(\overline{A}_i), \quad \overline{S}_2 = \sum_{1 \leq i_1 < i_2 \leq n} \mathrm{P}(\overline{A}_i \cap \overline{A}_j).$$

Then by applying the Poincare (sieve) formula we get for the differences between the probability value $\overline{P}$ and its second order Boole–Bonferroni bounds

$$\overline{P} - L_2 = \overline{S}_1 - \overline{S}_2 + \dots + (-1)^{n-1}\overline{S}_n - \frac{2}{k^*+1}\overline{S}_1 + \frac{2}{k^*(k^*+1)}\overline{S}_2,$$

and

$$\overline{P} - U_2 = -\overline{S}_2 + \dots + (-1)^{n-1}\overline{S}_n + \frac{2}{n}\overline{S}_2.$$

So the random variables

$$\nu_{L_2} = \begin{cases} 0, & \text{if } \mu \le 1 \\ \sum_{i=1}^{n} (-1)^{i-1}\binom{\mu}{i} - \frac{2}{k^*+1}\binom{\mu}{1} + \frac{2}{k^*(k^*+1)}\binom{\mu}{2} \; = \frac{1}{k^*(k^*+1)}(\mu - k^*)(\mu - k^* - 1), & \text{if } \mu \ge 2 \end{cases}$$

and

$$\nu_{U_2} = \begin{cases} 0, & \text{if } \mu \le 1 \\ \sum_{i=2}^{n} (-1)^{i-1}\binom{\mu}{i} - \frac{2}{n}\binom{\mu}{2} = \frac{1}{n}\,(\mu+n)\,(1-\mu)\,, & \text{if } \mu \ge 2 \end{cases}$$

have expected values $\overline{P} - L_2$ and $\overline{P} - U_2$. This way the transformed random variables $\nu_{L_2} + L_2$ and $\nu_{U_2} + U_2$ also have expected value $\overline{P}$.

6.3 Monte Carlo simulation of the difference between the true probability value and its Hunter–Worsley bound

The Hunter–Worsley upper bound for the probability $\overline{P} = \mathrm{P}(\overline{A}_1 \cup \overline{A}_2 \cup \cdots \cup \overline{A}_n)$ is defined as (see Prékopa's book [19]):

$$U_{HW} = \overline{S}_1 - \sum_{(i,j)\in T^*} \mathrm{P}(\overline{A}_i \cap \overline{A}_j) \ge \overline{P},$$

where T^* is the maximum weight spanning tree in the complete graph with nodes $\{1, \ldots, n\}$ and edges $\{\{i,j\}, 1 \le i,j \le n\}$ and to node i the probability $\mathrm{P}(\overline{A}_i)$ and to edge $\{i,j\}$ the probability $\mathrm{P}(\overline{A}_i \cap \overline{A}_j)$ is assigned.

Then by applying the Poincare formula we get for the difference between the probability value $\overline{P}$ and its Hunter–Worsley upper bound

$$\overline{P} - U_{HW} = -\overline{S}_2 + \overline{S}_3 - \ldots + (-1)^{n-1}\overline{S}_n + \sum_{(i,j)\in T^*} \mathrm{P}(\overline{A}_i \cap \overline{A}_j).$$

If λ denotes the number of those $\overline{A}_i \cap \overline{A}_j$, $(i,j) \in T^*$ events which occur in a random trial, then the random variable

$$\nu_{HW} = \begin{cases} 0, & \text{if } \mu \le 1 \\ \sum_{i=2}^{n} (-1)^{i-1}\binom{\mu}{i} + \lambda = 1 - \mu + \lambda, & \text{if } \mu \ge 2 \end{cases}$$

has expected value $\overline{P} - U_{HW}$ and so the transformed random variable $\nu_{HW} + U_{HW}$ also has expected value $\overline{P}$.

6.4 Determination of the final estimation with minimal variance

Let us chose the random variables ν_0, $\nu_{HW} + U_{HW}$ and $\nu_{L_2} + L_2$ and denote $\widehat{\overline{P}}_0, \widehat{\overline{P}}_1, \widehat{\overline{P}}_2$ the three different estimations based on these. All of these are unbiased estimations of the probability value $\overline{P}$. Let us estimate the empirical covariances of these estimations in a simulation procedure:

$$\widehat{C} = \begin{pmatrix} \widehat{c}_{00} & \widehat{c}_{01} & \widehat{c}_{02} \\ \widehat{c}_{01} & \widehat{c}_{11} & \widehat{c}_{12} \\ \widehat{c}_{02} & \widehat{c}_{12} & \widehat{c}_{22} \end{pmatrix}$$

If we introduce the new estimation

$$\widehat{\overline{P}} = w_0\widehat{\overline{P}}_0 + w_1\widehat{\overline{P}}_1 + w_2\widehat{\overline{P}}_2$$

where $w_0 + w_1 + w_2 = 1$, then it will be also an unbiased estimation of the probability value $\overline{P}$. As $\widehat{\overline{P}}$ has variance $w^T \widehat{C} w$, where $w^T = (w_0, w_1, w_2)$, therefore the coefficients w_0, w_1, w_2 resulting the minimal variance estimation can be determined by the solution of the nonlinear programming problem:

$$\min_{w_0+w_1+w_2=1} w^T \widehat{C} w.$$

As the gradient of $\mathbf{w}^T \widehat{C} \mathbf{w}$ equals to $2\mathbf{w}^T \widehat{C}$ it is easy to see that the unknown values of w_1, w_2, w_3, λ can be determined by the solution of the following linear equation system:

$$\begin{array}{ccccccccc} \widehat{c}_{00}w_0 & + & \widehat{c}_{01}w_1 & + & \widehat{c}_{02}w_2 & -\lambda & = 0, \\ \widehat{c}_{01}w_0 & + & \widehat{c}_{11}w_1 & + & \widehat{c}_{12}w_2 & -\lambda & = 0, \\ \widehat{c}_{02}w_0 & + & \widehat{c}_{12}w_1 & + & \widehat{c}_{22}w_2 & -\lambda & = 0, \\ w_0 & + & w_1 & + & w_2 & & = 1, \end{array} \tag{36}$$

representing the Karush-Kuhn-Tucker necessary condition.

6.5 Further Monte Carlo simulation algorithms

For the case of multivariate normal probability distribution there are other known Monte Carlo simulation algorithms, see Deák [3], [4] and Ambartzumian et al [1]. Gassmann [10] combined Szántai's general algorithm and Deák's algorithm into a hybrid algorithm. The efficiency of this algorithm was explored by Deák, Gassmann and Szántai in [5].

7 Handling a difficult constraint

We are going to work out an approximation scheme for the solution of the convex constrained problem (2). This scheme will consist of the solution of a sequence of problems of the form (1).

Let us consider the linear constraint set $A\boldsymbol{x} \leq \boldsymbol{b}$ of Problem (1). The last constraint of this set is $\boldsymbol{a}^r \boldsymbol{x} \leq b_r$, where $\boldsymbol{a}^r$ denotes the rth row of A, and b_r denotes the rth component of $\boldsymbol{b}$. Assume that this last constraint is a cost constraint, and let $\boldsymbol{c}^T = \boldsymbol{a}^r$ denote the cost vector. We are going to consider a parametric form of the cost constraint, namely, $\boldsymbol{c}^T \boldsymbol{x} \leq \boldsymbol{d}$, where $d \in \mathbb{R}$ is a parameter.

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

$$\min\ \phi(T\boldsymbol{x}) \quad \text{subject to} \quad \breve{A}\boldsymbol{x} \leq \breve{\boldsymbol{b}}, \quad \boldsymbol{c}^T \boldsymbol{x} \leq d, \tag{37}$$

with the parameter $d \in \mathbb{R}$. This parametric form of the unconstrained problem will be denoted by (1: $b_r = d$).

Let $\chi(d)$ denote the optimal objective value of Problem (37), as a function of the parameter d. This is obviously a monotone decreasing convex function. Let $\mathcal{I} \subset \mathbb{R}$ denote the domain over which the function is finite. We have either $\mathcal{I} = \mathbb{R}$ or $\mathcal{I} = [\underline{d}, +\infty)$ with some $\underline{d} \in \mathbb{R}$. Using the notation of the unconstrained problem, we can say that $\chi(d)$ is the optimum of (1: $b_r = d$) for $d \in \mathcal{I}$.

Coming to the constrained problem (2), we assume that the right-hand value π has been set by an expert, on the basis of preliminary experimental information. We may assume $\pi \in \chi(\mathcal{I})$. Let $d^\star \in \mathcal{I}$ be a solution of the equation $\chi(d) = \pi$, and let $l^\star(d)$ denote a linear support function to $\chi(d)$ at $d^\star$. Every decision maker would set a right-hand side π that satisfies

Assumption 15 *The support function $l^\star(d)$ has a significant negative slope, i.e., $l^{\star\prime} \ll 0$.*

It follows that the optimal objective value of (2) is $d^\star$.

We are going to find a near-optimal $\widehat{d} \in \mathcal{I}$ using an approximate version of Newton's method. – The idea of regulating tolerances in such a procedure occurs in the discussion of the Constrained Newton Method in

[14]. Based on the convergence proof of the Constrained Newton Method, a simple convergence proof of Newton's method was reconstructed in [8]. We are going to adapt the latter to the present case.

A sequence of unconstrained problems will be solved with increasing accuracy. In course of this procedure, we are going to build a single model $\phi_k(\boldsymbol{z})$ of the nonlinear objective $\phi(\boldsymbol{z})$. (I.e., k is ever increasing.) Given an iterate $d^\circ \in \mathcal{I}$, we can estimate $\chi(d^\circ)$ by solving the current model problem (7: $b_r = d^\circ$). Using the notation of the dual approach, let the optimum of the current model problem be $-\underline{\Psi} = -\phi_k^\star(\overline{u}) - \mu(\overline{u})$. According to (25), we have $-\underline{\Psi} \geq \chi(d^\circ) \geq -\underline{\Psi} - \overline{\mathcal{R}}$. If we can compute gradients of $\phi(\boldsymbol{z})$ then we can construct an upper bound of the gap $\overline{\mathcal{R}}$, as mentioned in Remark 7. If we work with gradient estimates then only a statistical estimation of final gap is possible, as mentioned in Remark 13. First we describe the approximation scheme for the former case. In order to handle gradient estimates, we work out a randomized version of the approximation scheme in Section 7.2.

7.1 A deterministic approximation scheme

In this section we consider the case when upper bounds can be constructed for the gap in the unconstrained problem. Let $d_0, d_1 \in \mathcal{I},\ d_0 < d_1 < d^\star$ be the starting iterates. – The sequence of the iterates will be strictly monotone increasing, and converging to $d^\star$ from below.

Near-optimality condition

Given a tolerance ϵ ($\pi \gg \epsilon > 0$), let $\widehat{d} \in \mathcal{I}$ be such that

$$\widehat{d} \leq d^\star \quad \text{and} \quad \chi(\widehat{d}\,) \leq \pi + \epsilon. \tag{38}$$

Let $\widehat{\boldsymbol{x}}$ be an optimal solution of (37: $d = \widehat{d}\,$). Then $\widehat{\boldsymbol{x}}$ is an ϵ-feasible solution of (2) with objective value $\widehat{d}$. Exact feasible solutions of (2) have objective values not less than $d^\star \geq \widehat{d}$.

Evaluation of $\chi(d_j)$

Given iterate $d_j \in \mathcal{I},\ d_j \leq d^\star$, we are going to include the unknown value $\chi(d_j)$ in a known interval whose length is comparable to $\chi(d_j) - \pi$. Namely, we are going to find an upper bound $\overline{\chi}_j$ such that

$$\overline{\chi}_j \ \geq\ \chi(d_j) \ \geq\ \overline{\chi}_j - \delta\,(\overline{\chi}_j - \pi), \tag{39}$$

where δ $(0 < \delta \ll \frac{1}{2})$ is a fixed tolerance.

Such $\overline{\chi}_j$ is found by approximately solving the problem (37: $d = d_j$) $\equiv$ (1: $b_r = d_j$). Let us use the column generation / cutting plane schemes described in previous sections. The optimum of the current model problem is always an upper bound for $\chi(d_j)$. Let us stop the column generation / cutting plane procedure when either of the following condition is satisfied:

$$\begin{aligned} &(i) \quad \overline{\chi}_j - \pi \ \leq \epsilon, \quad \text{or} \\ &(ii) \quad \overline{\mathcal{R}}_j \ \leq\ \delta\,(\overline{\chi}_j - \pi), \end{aligned} \tag{40}$$

where $\overline{\chi}_j$ will be an upper approximation of $\chi(d_j)$, and $\overline{\mathcal{R}}_j$ will measure the quality of this approximation.

Namely, let $\overline{\chi}_j$ be the optimum of the current model problem (7: $b_r = d_j$). We have

$$\overline{\chi}_j \ \geq\ \chi(d_j). \tag{41}$$

Let moreover $\overline{\mathcal{R}}_j$ be an upper estimate of the gap in the current model, i.e., of $\overline{\mathcal{R}}$. We have

$$\overline{\chi}_j - \chi(d_j) \ \leq\ \overline{\mathcal{R}}_j. \tag{42}$$

The tools described in Sections 3 and 4 allow us, in principle, to decrease $\overline{\mathcal{R}}_j$ below any tolerance.

If *(i)* occurs then, taking into account (41), $\widehat{d} := d_j$ satisfies the near-optimality condition (38).

If *(ii)* occurs then, taking into account (41)-(42), $\overline{\chi}_j$ satisfies (39), yielding a usable approximation of $\chi(d_j)$.

Finding successive iterates

Given $j \geq 1$, assume that we have evaluated $\chi(d_{j-1})$ and $\chi(d_j)$, as described in the previous section. We are going to find d_{j+1}. In order to do this, let us first consider the linear function $l_j(d)$ satisfying

$$l_j(d_{j-1}) := \overline{\chi}_{j-1} \ \geq \chi(d_{j-1}) \quad \text{and} \quad l_j(d_j) := \overline{\chi}_j - \delta\,(\overline{\chi}_j - \pi) \ \leq \chi(d_j), \tag{43}$$

where the inequalities follow from (39). Due to the convexity of $\chi(d)$ and to Assumption 15, the above linear function obviously has a negative slope $l'_j \leq l^{\star\prime} \ll 0$. Moreover $l_j(d) \leq \chi(d)$ holds for $d_j \leq d$.

The next iterate d_{j+1} will be the point where $l_j(d_{j+1}) = \pi$. Of course $d_j < d_{j+1} \leq d^\star$ follows from the observations above.

Convergence

Let the iterates $d_0, d_1, \ldots, d_s$ and the linear functions $l_1(d), \ldots, l_s(d)$ be as defined above. We assume that $s > 1$, and the procedure did not stop before step $(s+1)$. Then we have

$$\overline{\chi}_j - \pi > \epsilon \quad (j = 0, 1, \ldots, s). \tag{44}$$

To simplify notation, let us introduce $L_j(d) = l_j(d) - \pi \ \ (j = 1, \ldots, s)$. We transform (43) into

$$L_j(d_{j-1}) = \overline{\chi}_{j-1} - \pi \quad \text{and} \quad L_j(d_j) \ = \ (1-\delta)(\overline{\chi}_j - \pi) \qquad (j = 1, \ldots, s), \tag{45}$$

positivity of function values following from (44). Moreover, the derivatives satisfy

$$L'_j \ = \ l'_j \ \leq \ l^{\star\prime} \ \ll 0 \qquad (j = 1, \ldots, s) \tag{46}$$

due to the observations in the previous section.

Theorem 16 *We have*

$$\gamma^{(s-1)} \mathcal{K}\, L_1(d_1) \ \geq \ L_s(d_s), \tag{47}$$

where $0 < \gamma \ll 1$ *and* $\mathcal{K}$ *is a positive number of moderate magnitude.*

Proof. The following statements hold for $j = 1, \ldots, s-1$. From (45), we get

$$\frac{L_{j+1}(d_j)}{L_j(d_j)} \ = \ \frac{\overline{\chi}_j - \pi}{(1-\delta)(\overline{\chi}_j - \pi)} \ = \ \frac{1}{1-\delta}. \tag{48}$$

By definition, we have

$$L_j(d_j) + (d_{j+1} - d_j)\, L'_j \ = \ L_j(d_{j+1}) \ = 0.$$

It follows that $d_{j+1} - d_j = \frac{L_j(d_j)}{|L'_j|}$. Using this, we get

$$L_{j+1}(d_j) \ = \ L_{j+1}(d_{j+1}) + (d_j - d_{j+1})\, L'_{j+1} \ = \ L_{j+1}(d_{j+1}) + \frac{L_j(d_j)}{|L'_j|}\, |L'_{j+1}|.$$

Hence

$$\frac{L_{j+1}(d_j)}{L_j(d_j)} \ = \ \frac{L_{j+1}(d_{j+1})}{L_j(d_j)} + \frac{|L'_{j+1}|}{|L'_j|}. \tag{49}$$

From (48), we have

$$\frac{1}{1-\delta} \ = \ \frac{L_{j+1}(d_{j+1})}{L_j(d_j)} + \frac{|L'_{j+1}|}{|L'_j|} \ \geq \ 2\sqrt{\frac{L_{j+1}(d_{j+1})\,|L'_{j+1}|}{L_j(d_j)\,|L'_j|}}.$$

(This is the well-known inequality between means.) It follows that

$$\left(\frac{1}{2(1-\delta)}\right)^2 L_j(d_j)\,|L'_j| \;\geq\; L_{j+1}(d_{j+1})\,|L'_{j+1}|.$$

By induction, we get

$$\left(\frac{1}{2(1-\delta)}\right)^{2(s-1)} L_1(d_1)\,|L'_1| \;\geq\; L_s(d_s)\,|L'_s|. \tag{50}$$

Dviding both sides by $|L'_s|$, and substituting

$$\gamma := \left(\frac{1}{2(1-\delta)}\right)^2, \qquad \mathcal{K} := \frac{|L'_1|}{|l^{\star\prime}|} \geq \frac{|L'_1|}{|L'_s|},$$

we obtain (47). Here $\gamma \ll 1$ follows from the setting of the constant $0 < \delta \ll \frac{1}{2}$. $\mathcal{K}$ has a moderate magnitude according to (46), eventually due to Assumption 15.

7.2 A randomized version of the approximation scheme

In this section we consider the case when gradient estimates are used in solving the unconstrained problems. Only a statistical estimation of the gap $\overline{\mathcal{R}}$ is possible, on the basis of Assumption 10. Let $\overline{\mathcal{R}}_j$ denote our upper estimate, in accordance with the notation used for the deterministic scheme. We may underestimate the gap, meaning that (42) does not hold; and consequently the right-hand inequality of (39) may not hold. The probability of such an occurrence can be kept low. In such exceptional cases, $d_{j+1} > d^\star$ may occur.

Hence we need to check $\chi(d_{j+1})$ in the new iterate. If we can verify $\chi(d_{j+1}) > \pi$ then we can proceed as in the deterministic scheme. If we can verify $\chi(d_{j+1}) < \pi$ then we can just step back to the previous iterate d_j and re-estimate the gap, possibly with a higher reliability. (The process may involve tightening of the upper bound $\overline{\chi}_j$.)

It may happen that neither $\chi(d_{j+1}) > \pi$ nor $\chi(d_{j+1}) < \pi$ can be verified because $\chi(d_{j+1})$ is near to π. The latter fact can also be verified, and in this case d_{j+1} can be considered a near-optimal solution to (2), under mild additional assumptions on $\chi(.)$ and π.

8 Conclusion and discussion

In this paper we present the column-generation approach of [7] from a dual viewpoint, as a cutting-plane method. Moreover we propose a randomized version of this method. There is an important contrast between direct cutting-plane methods and the present approach. Direct cutting-plane methods for probabilistic functions are difficult to implement due to noise in gradient computation. Even a fairly accurate gradient may result a cut cutting into the epigraph of the objective function (especially in regions farther away from the current iterate). One either needs sophisticated tolerance handling to avoid cutting into the epigraph – see, e.g., [24], [16], [2] –, or else one needs a sophisticated convex optimization method that can handle cuts cutting into the epigraph – see [26]. (Yet another alternative, developed for a different type of problem, is perpetual adjustment of existing cuts to information revealed in course of the process; see [12].)

The present models are invulnerable to gradient computation errors. Noisy gradient estimates may yield iterates that do not improve much on our current models. But we retain a true inner approximation of the primal objective – or a true outer approximation of the dual objective –, provided objective values at new iterates are evaluated with appropriate accuracy. This feature facilitates the use of gradient estimates. Our randomized method bears a resemblance to the stochastic approximation family that goes back to [21] (see [17], [13] for recent forms).

The use of gradient estimates may substantially decrease total computational effort, even though a certain (moderate) accuracy *is* demanded in objective values. Computing a single component of a gradient

vector will involve an effort comparable to that of computing an objective value, e.g., in case of probability maximization under multivariate normal distribution of the random parameters.

The variance reduction Monte Carlo simulation procedure described in Section 6 was successfully applied in the solution of jointly probabilistic constrained stochastic programming problems, see [24]. The situation was similar to the present one; as the procedure progressed, higher and higher accuracy became necessary.

The approximation scheme of Section 7 consist of the solution of a sequence of problems of the form (1), i.e., minimization of a convex objective over polyhedra. Suppose the problems in this sequence are solved by the approximation approach described in previous sections. Then can we build a single model of the nonlinear objective $\phi(\boldsymbol{z})$. I.e., the model built in course of the solution of problem $\mathcal{P}_s$ can be used as a starting model for the solution of the successive problem $\mathcal{P}_{s+1}$.

Future work

Our motivation for dealing with a difficult function was a probabilistic function $F(\boldsymbol{z}) = \mathrm{P}(\boldsymbol{\xi} < \boldsymbol{z})$, where the random vector $\boldsymbol{\xi}$ has a logconcave distribution. The proposed approach can be extended to two-sided probabilities of the type

$$\mathrm{P}(\underline{\boldsymbol{z}} < \boldsymbol{\xi} < \overline{\boldsymbol{z}}), \tag{51}$$

where $\underline{\boldsymbol{z}}$ and $\overline{\boldsymbol{z}}$ are linear functions of the decision variables, i.e., we have $\underline{\boldsymbol{z}} = \underline{T}\boldsymbol{x} + \underline{\boldsymbol{t}}$ and $\overline{\boldsymbol{z}} = \overline{T}\boldsymbol{x} + \overline{\boldsymbol{t}}$ with appropriate matrices $\underline{T}, \overline{T}$ and vectors $\underline{\boldsymbol{t}}, \overline{\boldsymbol{t}}$. The Monte Carlo simulation procedures described in Section 6 can be applied by using

$$\overline{A}_i = \{\xi_i \le \underline{z}_i\} \cup \{\overline{z}_i \le \xi_i\} \quad (i = 1, \dots, n).$$

Van Ackooij, Henrion, Möller and Zorgati [25] developed gradient formulas for two-sided probabilities (51) in case of normal distributions. Analogous formulas can be developed for other multivariate probability distributions.

Acknowledgement

This research is supported by the EFOP-3.6.1-16-2016-00006 *"The development and enhancement of the research potential at Pallas Athena University"* project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] R. Ambartzumian, A. Der Kiureghian, V. Ohanian, and H. Sukiasian. Multinormal probability by sequential conditioned importance sampling: Theory and applications. *Probabilistic Engineering Mechanics*, 13:299–308, 1998.

[2] T. Arnold, R. Henrion, A. Möller, and S. Vigerske. A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. *Pacific Journal of Optimization*, 10:5–20, 2014.

[3] I. Deák. Three digit accurate multiple normal probabilities. *Numerische Mathematik*, 35:369–380, 1980.

[4] I. Deák. Computing probabilities of rectangles in case of multinormal distributions. *Journal of Statistical Computation and Simulation*, 26:101–114, 1986.

[5] I. Deák, H. Gassmann, and T. Szántai. Computing multivariate normal probabilities: a new look. *Journal of Statistical Computation and Simulation*, 11:920–949, 2002.

[6] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89:55–77, 2000.

[7] C.I. Fábián, E. Csizmás, R. Drenyovszki, W. van Ackooij, T. Vajnai, L. Kovács, and T. Szántai. Probability maximization by inner approximation. *Submitted to Acta Polytechnica Hungarica*, 2017.

[8] C.I. Fábián, K. Eretnek, and O. Papp. A regularized simplex method. *Central European Journal of Operations Research*, 23:877–898, 2015.

[9] C.I. Fábián, O. Papp, and K. Eretnek. Implementing the simplex method as a cutting-plane method, with a view to regularization. *Computational Optimization and Applications*, 56:343–368, 2013.

[10] H. Gassmann. Conditional probability and conditional expectation of a random vector. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 237–254. Springer-Verlag, Berlin, 1988.

[11] A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150, 1992.

[12] J.L. Higle and S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, volume 8 of *Nonconvex Optimization and Its Applications*. Springer, 1996.

[13] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133:365–397, 2012.

[14] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–147, 1995.

[15] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research and Management Science. Springer, 2008.

[16] J. Mayer. *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*. Gordon and Breach Science Publishers, 1998.

[17] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

[18] A. Prékopa. Dual method for a one-stage stochastic programming problem with random RHS obeying a discrete probability distribution. *ZOR - Methods and Models of Operations Research*, 34:441–461, 1990.

[19] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, 1995.

[20] A. Prékopa, B. Vizvári, and T. Badics. Programming under probabilistic constraint with discrete random variable. In F. Giannesi, T. Rapcsák, and S. Komlósi, editors, *New Trends in Mathematical Programming*, pages 235–255. Kluwer, Dordrecht, 1998.

[21] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[22] A. Ruszczyński. *Nonlinear Optmization*. Princeton University Press, 2006.

[23] T. Szántai. *Numerical Evaluation of Probabilities Concerning Multidimensional Probability Distributions, Thesis*. Hungarian Academy of Sciences, Budapest, 1985.

[24] T. Szántai. A computer code for solution of probabilistic-constrained stochastic programming problems. In Y.M. Ermoliev and R.J.-B. Wets, editors, *Numerical Techniques for Stochastic Optimization*, pages 229–235. Springer-Verlag, Berlin, 1988.

[25] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research*, 71:535–549, 2010.

[26] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24:733–765, 2014.