

A simulation-based approach to two-stage stochastic programming with recourse

Alexander Shapiro *, Tito Homem-de-Mello ¹

*School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, GA 30332-0205, USA*

Received 10 March 1996; revised manuscript received 9 March 1997

Abstract

In this paper we consider stochastic programming problems where the objective function is given as an expected value function. We discuss Monte Carlo simulation based approaches to a numerical solution of such problems. In particular, we discuss in detail and present numerical results for two-stage stochastic programming with recourse where the random data have a continuous (multivariate normal) distribution. We think that the novelty of the numerical approach developed in this paper is twofold. First, various variance reduction techniques are applied in order to enhance the rate of convergence. Successful application of those techniques is what makes the whole approach numerically feasible. Second, a statistical inference is developed and applied to estimation of the error, validation of optimality of a calculated solution and statistically based stopping criteria for an iterative algorithm. © 1998 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Two-stage stochastic programming with recourse; Monte Carlo simulation; Likelihood ratios; Variance reduction techniques; Confidence intervals; Hypotheses testing; Validation analysis; Nonlinear programming

1. Introduction

In many practical situations one is required to solve optimization problems which are subject to uncertainty. There are various ways to model an uncertainty (incomplete information, data variability, randomness, etc.) which lead to different formulations of the associated optimization problems. In this paper we focus on a particular approach to such problems, which is based on simulation (Monte Carlo) techniques, and apply it to a specific class of problems.

Consider the optimization problem

$$\min_{x \in S} E\{f(x, \zeta)\} \tag{1}$$

* Corresponding author. E-mail: ashapiro@isye.gatech.edu.

¹ Supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brasília, Brazil, through a Doctoral Fellowship under grant 200595/93-8.

of minimization of the expected value function $E\{f(x, \zeta)\}$ over a set $S \subset \mathbb{R}^m$. We assume that the set S is deterministic and is given explicitly by linear or nonlinear constraints and that ζ is a random vector whose distribution is known. In realistic applications, and especially when the random vector ζ has a large dimensionality, it is typically impossible to calculate the expected value $E\{f(x, \zeta)\}$ in a closed form and hence numerical approximations are required. The basic idea of an approach that we discuss in this paper is based on Monte Carlo techniques and is quite simple. A random sample Z_1, \dots, Z_N of independent replications of the random vector ζ is generated and consequently the expected value function is approximated by the average function.

$$\hat{f}_N(x) = N^{-1} \sum_{i=1}^N f(x, Z_i). \quad (2)$$

The idea of generation of a random sample and consequent approximation of the expectation by the corresponding average is not new, of course, and is the heart of the Monte Carlo method. Somewhat recently Monte Carlo simulation based numerical techniques started to attract attention in stochastic programming community. We can mention in that respect the stochastic subgradient (stochastic quasigradient) methods [1,2], and approaches developed in [3,4]. In this paper we consider situations when, for a generated sample Z_1, \dots, Z_N , the value, first and possibly second order derivatives of the average function $\hat{f}_N(x)$ can be calculated and hence deterministic algorithms of nonlinear programming can be applied to minimization of $\hat{f}_N(x)$ over S (cf. [5,6]). We discuss the problem and report a numerical experience for a particular class of stochastic programs, namely two-stage stochastic programs with recourse. Although the ideas discussed here are concentrated on two-stage stochastic programming with recourse, we believe that some of them can be applied to a wider range of problems.

We think that the novelty of numerical techniques developed in this paper is two-fold. First, it was possible to apply various variance reduction techniques which enhanced the rate of convergence and in fact made the whole approach numerically feasible. Second, a statistical inference was developed and applied to estimation of the error, validation of optimality of a calculated solution and statistically based stopping criteria for an iterative algorithm.

2. Two-stage recourse problem

In this section we discuss some basic ideas applied to two-stage stochastic programming with recourse. Stochastic programs with recourse were introduced in the fifties by Dantzig [7] and Beale [8]. For more recent discussions of this class of problems and extended bibliography see e.g., [9,2,10,11]. Consider the optimization problem

$$\min_{x \in S} c^T x + E\{Q(x, \omega)\}, \quad (3)$$

where

$$Q(x, \omega) = \inf \{ q^T y : Wy = h(\omega) - T(\omega)x, y \geq 0 \}. \tag{4}$$

Here $h = h(\omega)$ is an $n \times 1$ random vector and $T = T(\omega)$ is an $n \times m$ random matrix defined on a probability space (Ω, \mathcal{F}, P) . We make the following (stochastic) assumptions throughout the paper: (i) h and T are independent, and (ii) the distribution of the random vector h has a probability density function (pdf) $p(\cdot)$.

With few exceptions (e.g. [3,4]), existing numerical methods for solution of (3) are based on deterministic techniques which deal with a *finite* number of realizations of the corresponding random variables. This, in turn, requires discretization of the underlying probability measures (distributions) in case these distributions are continuous. In many situations even a reasonably moderate number of such realizations results in huge linear programs which cannot be solved even by modern computers.

Note that the function $Q(x, \omega)$ can be written in the form $Q(x, \omega) = G(h(\omega) - T(\omega)x)$, where

$$G(z) = \inf \{ q^T y : Wy = z, y \geq 0 \}. \tag{5}$$

By duality arguments (cf. [12]) the function $G(\cdot)$ can be represented in the form

$$G(z) = \sup \{ \xi^T z : W^T \xi \leq q \}. \tag{6}$$

For the sake of simplicity we assume that: (i) for every vector z the system $Wy = z, y \geq 0$, has a solution (the recourse is complete), and (ii) the system $W^T \xi \leq q$ has a solution (dual feasibility). Under these assumptions, $G(\cdot)$ is a finite valued, piecewise linear convex function. We also assume that the expectation $E\{Q(x, \omega)\}$ exists for all $x \in S$.

Suppose now that a random sample $(h_1, T_1), \dots, (h_N, T_N)$, of i.i.d. (independent identically distributed) realizations of $(h(\omega), T(\omega))$ is generated. Then the expected value function $g(x) = E\{Q(x, \omega)\}$ can be estimated by the sample average function

$$\hat{g}_N(x) = N^{-1} \sum_{i=1}^N G(h_i - T_i x), \tag{7}$$

and consequently the program (3) can be approximated by

$$\min_{x \in S} c^T x + \hat{g}_N(x). \tag{8}$$

One can solve problem (8) by using deterministic methods of nonlinear programming and then to use its optimal solution as an approximation of the optimal solution of the original problem (3). This approach, known as a *stochastic counterpart* (SC) method or a *sample-path* optimization, has been discussed and analyzed, for example, in [13,5,14,6]. Of course, an implementation of that idea requires specification of a particular algorithm which is used for solving the approximating problem (8).

Notice that $G(\cdot)$ is a piecewise linear, nondifferentiable convex function. It follows that $\hat{g}_N(\cdot)$ is also a piecewise linear, nondifferentiable convex function. Nevertheless,

we can compute a subgradient of $\hat{g}_N(\cdot)$ as follows. Let $\partial G(z)$ denote the subdifferential of $G(\cdot)$ at z . By (6) we have that

$$\partial G(z) = \arg \max \{ \xi^T z : W^T \xi \leq q \}.$$

Furthermore, by standard subdifferential calculus we have that

$$\partial_x G(h - Tx) = -T^T \partial G(h - Tx). \tag{9}$$

Consequently we have that a subgradient of $\hat{g}_N(\cdot)$ is given by

$$\nabla \hat{g}_N(x) = -N^{-1} \sum_{i=1}^N T^T \nabla G(h_i - T_i x), \tag{10}$$

where $\nabla G(z)$ denotes a subgradient of $G(\cdot)$ at z , which in turn is given by any optimal solution of the linear programming problem

$$\begin{aligned} \max \quad & \xi^T z \\ \text{s.t.} \quad & W^T \xi \leq q. \end{aligned}$$

It is important to observe that the expected value function $g(\cdot) := E\{Q(\cdot, \omega)\}$ is differentiable and that, for any given x , $\nabla \hat{g}_N(x)$ is a consistent estimator of $\nabla g(x)$, i.e., $\nabla \hat{g}_N(x)$ converges w.p.1. to $\nabla g(x)$ as $N \rightarrow \infty$. In order to see that, note initially that convexity of the function $G(\cdot)$ implies that $g(\cdot)$ is also convex and its subdifferential can be taken inside the expected value (see [15] for details). That is,

$$\partial g(x) = E \left\{ \partial_x G(h - Tx) \right\} = E \left\{ -T^T \partial G(h - Tx) \right\}. \tag{11}$$

Furthermore, by Rademacher theorem, the set of points where $G(\cdot)$ is not differentiable has Lebesgue measure zero. It follows that $\partial G(h - Tx)$ is a singleton with probability one (since we assume that h has a density and h and T are independent). Consequently $g(x)$ is differentiable at x and, by the Strong Law of Large Numbers, the estimator $\nabla \hat{g}_N(x)$, defined in (10), is a consistent estimator of $\nabla g(x)$.

It should be noted that second order derivatives of the expected value function $g(\cdot)$ cannot be taken inside the expected value. In fact second order derivatives of $\hat{g}_N(x)$ are zeros whenever they exist. An alternative approach to estimation of $g(x)$, which also allows an estimation of its first and second order derivatives, is to make a change-of-variables transformation and consequently to apply the *likelihood ratio* (LR) method (cf. [16,6]) as follows. Note first that we can write the expected value function $g(x)$ in the form

$$g(x) = E_T \left\{ E_h [Q(x, h, T) | T] \right\} = E_T \left\{ \int_{\mathbb{R}^n} G(\eta - Tx) p(\eta) \, d\eta \right\} \tag{12}$$

and by making the transformation $y = \eta - Tx$,

$$g(x) = E_T \left\{ \int_{\mathbb{R}^n} G(y) p(y + Tx) \, dy \right\}. \tag{13}$$

It follows that (cf. [6]) we can represent $g(x)$ in the form

$$g(x) = E_T \left\{ \int_{\mathbb{R}^n} G(y) \frac{p(y + Tx)}{p_0(y)} p_0(y) dy \right\} = E_T \left\{ E_{p_0} \{ G(Y)L(Y, T, x) \} \right\}, \tag{14}$$

where $p_0(y)$ is a chosen pdf, referred to as the dominating pdf, Y is a random vector whose distribution is determined by the dominating pdf $p_0(y)$ and

$$L(y, \tau, x) = \frac{p(y + \tau x)}{p_0(y)} \tag{15}$$

is the so-called LR function.

In order to simplify the presentation we assume subsequently that only the vector h is random while the matrix T is *fixed* (deterministic). In that case the LR function can be written in the form

$$L(y, x) = \frac{p(y + Tx)}{p_0(y)}. \tag{16}$$

Now let Y_1, \dots, Y_N be a random sample, where Y_i are generated from the chosen pdf $p_0(y)$. Then, because of (14), we can estimate $g(x)$ by the average function

$$\tilde{g}_N(x) = N^{-1} \sum_{i=1}^N G(Y_i)L(Y_i, x) = N^{-1} \sum_{i=1}^N w_i p(Y_i + Tx), \tag{17}$$

where $w_i = G(Y_i)/p_0(Y_i)$. Note that the approximating function $\tilde{g}_N(\cdot)$ is given explicitly provided that values $G(Y_1), \dots, G(Y_N)$ are calculated and the pdf $p(\cdot)$ is given in a closed form. For example, if h has a multivariate normal distribution, then $p(\cdot)$ can be written in the form

$$p(\eta) = k e^{-\frac{1}{2}(\eta - \mu)^T \Sigma^{-1}(\eta - \mu)}, \tag{18}$$

where μ and Σ are the mean vector and the covariance matrix of h , respectively, and k is a normalization constant.

Note also that the function $\tilde{g}_N(\cdot)$ is smooth, say twice continuously differentiable, if the pdf $p(\cdot)$ is smooth. Then under standard regularity conditions, given a point x , the gradient $\nabla \tilde{g}_N(x)$ and the Hessian matrix $\nabla^2 \tilde{g}_N(x)$ provide consistent estimates of the gradient $\nabla g(x)$ and the Hessian matrix $\nabla^2 g(x)$ of the expected value function $g(x)$, respectively (see [6] for details). Moreover, it is possible to reduce the variance of the obtained estimates by controlling the choice of the dominating pdf $p_0(\cdot)$ (cf. [17,6]). Also the required values $G(Y_i)$, $i = 1, \dots, N$, can be calculated independently of each other which can be convenient for parallel computation.

3. Computational issues

Consider the estimator $\tilde{g}_N(x)$, defined in (17), and the corresponding program

$$\min_{x \in S} c^T x + \tilde{g}_N(x) \tag{19}$$

giving an approximation of the program (3). The above program is different from (8) in that the ‘straightforward’ estimator $\hat{g}_N(x)$ is replaced by the LR estimator $\tilde{g}_N(x)$. An advantage of the LR estimator is that once the sample Y_1, \dots, Y_N is generated from the chosen (and fixed) pdf $p_0(y)$, and the values $G(Y_i)$ are computed, the LR average function $\tilde{g}_N(\cdot)$ is given *explicitly* through the LR’s $L(Y_i, \cdot)$. Consequently (19) becomes a smooth nonlinear (deterministic) programming problem. It is tempting then to try to solve the obtained problem (19) and to use its optimal solution as an estimator of the optimal solution of the expected value problem (3). Unfortunately such an approach did not work well in the present case. In order to see why let us make a quick analysis of the above problem.

The expected value function $g(x)$ in convex irrespective of the underlying distribution while the function $\tilde{g}_N(x)$ is given through the corresponding pdf $p(\cdot)$ and can be nonconvex. It is possible to show that $\nabla^2 \tilde{g}_N(x)$ converges w.p.1. to $\nabla^2 g(x)$ *uniformly* on any compact set $C \subset \mathbb{R}^m$ (e.g. [6]). It is also possible to show that the Hessian matrices $\nabla^2 g(x)$ are positive definite, provided the random vector h has a positive valued density function and the matrix T has full column rank [18]. It follows that w.p.1. for N large enough, the Hessian matrices $\nabla^2 \tilde{g}_N(x)$ are also positive definite, and hence $\tilde{g}_N(x)$ in convex, on C . However, such mathematical statements should be taken cautiously. Since the pdf $p(\eta) \rightarrow 0$ as $\eta \rightarrow \infty$, the approximating function $\tilde{g}_N(x)$ also tends to zero, even if $g(x) \rightarrow \infty$ as $x \rightarrow \infty$. Therefore $\tilde{g}_N(x)$ cannot be convex and cannot give a good approximation of $g(x)$ on the whole space \mathbb{R}^m . We come here to the concept of a (stochastic) *trust region*, that is a region where $\tilde{g}_N(x)$ can be trusted to give a reasonably good approximation of $g(x)$. It turns out that, typically, such a trust region is too small to be useful for optimization purposes.

Suppose, for instance, that h has a multivariate normal distribution (see (18)) with mean μ and covariance matrix Σ and that the pdf $p_0(\cdot)$ is also multivariate normal with the same covariance matrix and mean μ_0 . Then the variance of $\tilde{g}_N(x)$ is given by

$$\sigma_N^2(x) = \text{var} \left\{ \tilde{g}_N(x) \right\} = N^{-1} \left[E_{p_0} \left\{ G(Y)^2 L(Y, x)^2 \right\} - g(x)^2 \right] \tag{20}$$

with (see [6, p. 52])

$$E_{p_0} \left\{ G(Y)^2 L(Y, x)^2 \right\} = E_{\mu_0} \left\{ L(Y, x)^2 \right\} E_{\mu_0+2\delta} \left\{ G(Y)^2 \right\} = e^{\delta^T \Sigma^{-1} \delta} E_{\mu_0+2\delta} \left\{ G(Y)^2 \right\}, \tag{21}$$

where $\delta = \mu(x) - \mu_0$ and $\mu(x) = \mu - Tx$. Formulas (20) and (21) show that the variance of $\tilde{g}_N(x)$ grows exponentially with δ whenever the term $E_{\mu_0+2\delta} \{G(Y)^2\}$ is bounded from below by a positive constant. It follows that the trust region tends to be small, thus preventing long steps in the process of minimization of $\tilde{g}_N(x)$ and therefore making direct solving of (19) not feasible from a practical standpoint.

Nevertheless the LR estimator $\tilde{g}_N(x)$ can be useful in several respects. First, the function $\tilde{g}_N(\cdot)$ is smooth, provided the pdf $p(\cdot)$ is smooth, and hence its second order derivatives can be used in order to estimate the corresponding second order derivatives of $g(\cdot)$. Second, $\tilde{g}_N(\cdot)$ can be employed in conjunction with some variance reduction techniques (see Section 6.2).

A conceptual idea of the algorithm, implemented in this paper, can be described now as follows. Given a current iteration point x^k , a random sample is generated from a current pdf $p_0^k(\cdot)$ and then a few steps of a chosen algorithm (e.g. sequential quadratic minimization) are applied to an approximating nonlinear (deterministic) programming problem. In this respect both estimators $\hat{g}_N(\cdot)$ and $\tilde{g}_N(\cdot)$, and their derivatives, are used in order to improve the accuracy of approximation and to enhance rate of convergence. Then, for the next iteration point x^{k+1} , a new random sample is generated (possibly of a larger size and from a different density $p_0^{k+1}(\cdot)$), the approximating program is updated and a few steps of the algorithm are applied to the updated program, etc. As we shall see in Sections 4 and 5, such resampling is essential to ensure *independence* (in the probabilistic sense) between the estimators of $\nabla g(x^k)$ and $\nabla g(x^{k+1})$ (conditionally on the value of x^{k+1}), which is a required condition for an application of the statistical optimality tests and implementation of the stopping rules described there. The latter argument also suggests the use of resampling only at last iterations of the algorithm. A framework for proving convergence (with probability one) of such an algorithm is discussed in [19].

3.1. Increasing sample sizes

An important issue concerns the *size* of the sample used to compute the estimators (7) and (17) as well as their derivatives. Numerical experiments indicate that well controlled choice of the sample sizes can significantly reduce the computational time and improve the accuracy of obtained solutions. At first steps of the algorithm, when the current iteration point is far from the optimal, there is no need to have high precision estimates. On the other hand, at each iteration the employed estimates, of the expected value function and its derivatives, should be accurate enough in order for the algorithm to proceed in significant improvement of a current solution. That is, at each iteration, on one hand we would like to use a small sample in order to save computational time, on the other hand the sample should be large enough in order for the algorithm to proceed. The required compromise can be achieved by techniques of statistical testing, of the employed estimates of the gradient of $g(x)$, which we describe now.

Consider a feasible point $x \in S$, representing a current iteration point of the algorithm. Let $\gamma_N(x)$ be an estimator of the gradient $\nabla g(x)$ such that $\gamma_N(x) \rightarrow \nabla g(x)$ w.p.l., as $N \rightarrow \infty$, and $\gamma_N(x)$ has approximately (asymptotically) a multivariate normal distribution with the mean vector $\nabla g(x)$ and a covariance matrix Ω_N . For example, if the gradient of $g(x)$ is estimated by the gradient $\gamma_N(x) := \nabla \hat{g}_N(x)$ or $\gamma_N(x) := \nabla \tilde{g}_N(x)$ of the corresponding average function, then asymptotic normality

of $\gamma_N(x)$ follows by the Central Limit Theorem. Note that in this case the covariance matrix Ω_N can be estimated by S_N/N , where S_N is the sample covariance matrix. It follows that the random variable

$$N(\gamma_N(x) - \nabla g(x))^T S_N^{-1} (\gamma_N(x) - \nabla g(x))$$

has approximately (asymptotically) a chi-square distribution with m degrees of freedom, where m is the dimensionality of x and of $\gamma_N(x)$ (see, e.g., [20]). Consequently an (approximate) $100(1 - \alpha)\%$ confidence region for $\nabla g(x)$ is given by the following ellipsoid

$$E_r(x) := \left\{ z \in \mathbb{R}^m: (z - \gamma_N(x))^T S_N^{-1} (z - \gamma_N(x)) \leq r \right\},$$

where $r := \chi_m^2(\alpha)/N$ and $\chi_m^2(\alpha)$ is the constant corresponding to the significance level α .

The size of the above ellipsoid $E_r(x)$ is determined by the constant r , which in turn is inversely proportional to the sample size N . Suppose now that the feasible set S is defined by a finite number of linear constraints. Then our criterion for the choice of N is to find an ellipsoid of maximal size $r = r_N^*$ satisfying the following property. Consider the null space \mathcal{L} of the matrix generated by constraints defining the set S which are active at the point x . This linear space is contained in the set of feasible directions tangent to S at the point x . Let P be the orthogonal projection onto \mathcal{L} . The property that we want the above confidence region (ellipsoid) $E_r(x)$ to satisfy is that for any $z \in E_r(x)$, vector $P(c + z)$ forms an acute angle with $P(c + \gamma_N(x))$. Such choice of the sample size N guarantees that, with given confidence $100(1 - \alpha)\%$, the projection $P(c + \nabla g(x))$, of the gradient of the objective function of the problem (3), forms an acute angle with $P(c + \gamma_N(x))$, and hence the estimated direction, at least approximately, is a direction of descent for the ‘true’ problem (3).

We proceed now as follows. We need to compute $r_N^* = \max\{r: \psi(r) \geq 0\}$, where

$$\psi(r) = \min_{z \in E_r(x)} a^T P(c + z),$$

and $a := P(c + \gamma_N(x))$. By solving the KKT conditions for the above problem it can be shown that

$$\psi(r) = a^T a - r^{1/2} (a^T S_N a)^{1/2},$$

and hence the optimal r_N^* is

$$r_N^* = (a^T a)^2 (a^T S_N a)^{-1}. \tag{22}$$

Consequently the new sample size is computed as

$$N' = \max \left\{ \frac{\chi_m^2(\alpha)}{r_N^*}, N \right\}. \tag{23}$$

It should be noted that when N is not large enough, S_N can be a poor estimator of the corresponding covariance matrix and hence the above computation can lead to a very large value of the new sample size N' . For an actual implementation we suggest that the ‘jump’ from N to N' should be limited by a constant factor, say ten times.

4. Validation analysis

Suppose that we are given a point x^* which is suggested as an approximation of an optimal solution x_0 of the program (1). Can we evaluate the quality of this approximation? Closely related to this question is a choice of stopping criteria for a considered algorithm. In this section we discuss some statistical tests for validation of optimality of the solution x^* (cf. [21]). The discussion of this section is quite general and is not restricted to the considered example of stochastic programming with recourse.

Suppose that the expected value function $f(x) := E\{f(x, \zeta)\}$ is differentiable at the point x^* . Also, assume that the feasible set S is defined by constraints as follows

$$S = \left\{ x \in \mathbb{R}^m : c_i(x) = 0, i = 1, \dots, k, c_i(x) \geq 0, i = k + 1, \dots, l \right\}, \tag{24}$$

where $c_i(x)$ are (deterministic) continuously differentiable functions. By the first order (KKT) optimality conditions we have that if x_0 is an optimal solution of the problem (1), then (under a constraint qualification) there exist Lagrange multipliers λ_i such that $\lambda_i \geq 0, i \in J(x_0)$, and

$$\nabla f(x_0) - \sum_{i \in I(x_0)} \lambda_i \nabla c_i(x_0) = 0, \tag{25}$$

where $J(x) = \{i : c_i(x) = 0, i = k + 1, \dots, l\}$ denotes the index set of inequality constraints active at x and $I(x) = \{1, \dots, k\} \cup J(x)$. Consider the polyhedral cone

$$C(x) = \left\{ z \in \mathbb{R}^m : z = \sum_{i \in I(x)} \alpha_i \nabla c_i(x), \alpha_i \geq 0, i \in J(x) \right\}. \tag{26}$$

Then the KKT optimality conditions (25) can be written in the form $\nabla f(x_0) \in C(x_0)$.

Suppose now that the gradient $\nabla f(x^*)$ can be estimated by a (random) vector $\gamma_N(x^*)$ such that $\gamma_N(x^*) \rightarrow \nabla f(x^*)$ w.p.1., as $N \rightarrow \infty$, and $\gamma_N(x^*)$ has (asymptotically) a multivariate normal distribution with the mean vector $\nabla f(x^*)$ and a covariance matrix Ω_N . By using the estimator $\gamma_N(x^*)$, we can test the hypothesis:

$$H_0 : \nabla f(x^*) \in C(x^*) \text{ against the alternative, } H_1 : \nabla f(x^*) \notin C(x^*). \tag{27}$$

In order to test the (optimality-conditions) hypothesis H_0 we suggest the following procedures. Suppose that the covariance matrix Ω_N is nonsingular, and hence is positive definite, and that a consistent estimator $\hat{\Omega}_N$ of Ω_N is available. Then we define our first test statistic as follows

$$T_1 = \min_{z \in C(x^*)} (\gamma_N(x^*) - z)^T \hat{\Omega}_N^{-1} (\gamma_N(x^*) - z). \tag{28}$$

This statistic is an asymptotic analogue of Hotelling’s test statistic which is used in multivariate analysis (e.g. [20]). It is possible to show (see, e.g. [20]) that if all Lagrange multipliers corresponding to the inequality constraints active at x^* are positive (*strict complementarity condition*), then the test statistic T_1 has approximately (asymptotically) a noncentral chi-square distribution with $m - s$ degrees of freedom, where

$$s = \text{card}(I(x^*)) = k + \text{card}(J(x^*)),$$

and the noncentrality parameter

$$\kappa = \min_{z \in C(x^*)} (\nabla f(x^*) - z)^T \Omega_N^{-1} (\nabla f(x^*) - z). \tag{29}$$

In particular, under H_0 we have that $\kappa = 0$ and hence the null distribution of T_1 is central chi-square with $m - s$ degrees of freedom. Therefore for a calculated value T_1 of the test statistic we can calculate the p -value, that is $p = \text{Prob}\{\chi_{m-s}^2 \geq T_1\}$. This p -value gives an indication of the quality of the suggested solution x^* with respect to the stochastic precision. A large (close to one) p -value means that such precision was reached, so the algorithm cannot proceed further, whereas a small (close to zero) p -value indicates that either the current solution is far from the optimal or the deterministic error starts to dominate. Such test should then be combined with other criteria, for instance a test of significance of reduction in the value of the function, as described in Section 5.

An alternative test statistic can be written in the form

$$T_2 = \min_{z \in C(x^*)} (\gamma_N(x^*) - z)^T (\gamma_N(x^*) - z). \tag{30}$$

This test statistic is simply the squared Euclidean distance from $\gamma_N(x^*)$ to the cone $C(x^*)$. From a numerical point of view, T_2 is more convenient than T_1 since it does not involve inversion of the covariance matrix $\hat{\Omega}_N$, which in some cases can be nearly singular (ill-conditioned). If the strict complementarity condition holds, then under H_0 the asymptotic distribution of T_2 is given by the distribution of a weighted sum of chi-square variables (see e.g. [22]). That is, T_2 has approximately the same distribution as the distribution of the random variable $\sum_{i=1}^m \alpha_i X_i$, where X_1, \dots, X_m are independent random variables, each having a chi-square distribution with one degree of freedom, α_i are eigenvalues of the matrix $P\Omega_N$, P is the projection matrix $P = I_m - A(A^T A)^{-1} A^T$ and A is the $m \times s$ matrix whose columns are formed from the gradient vectors $\nabla c_i(x^*)$, $i \in I(x^*)$. In this case, the p -value can be approximately computed by replacing the distribution of $\sum_{i=1}^m \alpha_i X_i$ by that of $c\chi_v^2 + b$, where c, v and b are chosen in such a way that $c\chi_v^2 + b$ and $\sum_{i=1}^m \alpha_i X_i$ have the same first three moments. This type of procedure is called *Pearson’s approach*, see [22] for details.

Let us make the following remarks. By accepting (i.e. by failing to reject) H_0 hypothesis we do not claim that H_0 actually holds, i.e. that x^* is an exact optimal

solution of (1). Accepting of H_0 simply means that, given the available stochastic precision, we cannot separate x^* from x_0 . Together with the corresponding confidence intervals, the above p -value gives a good indication of the quality of a calculated approximation x^* of the ‘true’ optimal solution x_0 .

It should be remembered that the mentioned null distributions of the test statistics T_1 and T_2 are *asymptotic*. Consider, for example, cases where $s = m$, i.e. the number of equality and active inequality constraints is the same as the number of decision variables, and suppose that the strict complementarity condition holds. Then if $\gamma_N(x^*)$ is sufficiently close to $\nabla f(x^*)$, we have that $\gamma_N(x^*) \in C(x^*)$, since in that case $\nabla f(x^*) \in \text{int } C(x^*)$. Therefore in such cases T_1 and T_2 should be zero with probability tending to one, as the sample size tends to infinity, and hence asymptotically T_1 and T_2 are considered to be identically zero. Of course, for a finite sample size these statistics are not identically zero. Nevertheless even in such extreme cases statistics T_1 and T_2 usually give a good indication for a quality of a suggested solution.

Note that without the strict complementarity condition the (asymptotic) distributions of T_1 and T_2 are more involved. In such cases asymptotic analysis of T_1 is related to the so-called chi-bar-squared distributions (see [23,24]). In general this problem requires further investigation.

5. Statistical inference

In this section we briefly discuss some asymptotic results associated with the approximating program (19). The same considerations hold for problem (8) as well. Let v_N be the optimal value and \hat{x}_N be an optimal solution of the program (19) and let v_0 be the optimal value and x_0 be the optimal solution of the limiting (expected value) program (3) (x_0 is assumed to be unique). We discuss subsequently statistical properties of v_N and \hat{x}_N considered as estimators of their ‘true’ counterparts v_0 and x_0 , respectively. We assume throughout this section that v_N and \hat{x}_N are consistent estimators in the sense that with probability one $v_N \rightarrow v_0$ and $\hat{x}_N \rightarrow x_0$ as $N \rightarrow \infty$. Simple conditions ensuring such consistency are that \tilde{g}_N converge w.p.1. to g uniformly on S (i.e. the uniform version of the strong Law of Large Numbers holds) and that \hat{x}_N stay w.p.1. in a bounded subset of S (see, e.g. [6]).

By the Central Limit Theorem we have that for any fixed $x \in S$,

$$N^{1/2} \left[\tilde{g}_N(x) - g(x) \right] \Rightarrow N(0, \sigma^2(x)), \tag{31}$$

where ‘ \Rightarrow ’ stands for convergence in distribution and $N(0, \sigma^2)$ denotes a normal distribution with mean zero and variance σ^2 . Note that the sample, which is used for generating the approximation function, can be also used for calculation of an estimator $\hat{\sigma}_N^2(x)$ (sample variance) of $\sigma^2(x)$.

Consider the optimal value v_N . Under mild regularity conditions the following asymptotic result holds (cf. [6, pp. 266–268, 25]),

$$N^{1/2} \left(v_N - v_0 \right) \Rightarrow N \left(0, \sigma^2(x_0) \right). \tag{32}$$

The corresponding asymptotic variance $\sigma^2(x_0)$ can be estimated by $\tilde{\sigma}_N^2 = \hat{\sigma}_N^2(\hat{x}_N)$. Consequently a *confidence interval* for the ‘true’ optimal value v_0 can be written in the form

$$\left[v_N - z_{\alpha/2} N^{-1/2} \tilde{\sigma}_N, v_N + z_{\alpha/2} N^{-1/2} \tilde{\sigma}_N \right],$$

where the constant $z_{\alpha/2}$ is related to a chosen confidence. For example, $z_{\alpha/2} = 1.96$ approximately corresponds to the probability (confidence) of 95%.

This confidence interval is especially convenient because of its simplicity and generality. For a current estimate \hat{x}_N of x_0 , this interval is simply the confidence interval for the value $f(\hat{x}_N)$ of the true objective function (recall that here $f(x) = c^T x + g(x)$). It is also closely related to the following question. Suppose that we are given two iteration points x^k and x^{k+1} in S . Is it possible to verify that x^{k+1} is a significantly better solution of (3) than x^k ? We can approach this problem by testing the hypothesis:

$$H_0: f(x^k) = f(x^{k+1}) \text{ against the alternative, } H_1: f(x^k) > f(x^{k+1}).$$

For that purpose the following asymptotic analogue of the standard *t*-test can be applied. Suppose that two *independent* samples of sizes N_1 and N_2 are generated and $g(x^k)$ and $g(x^{k+1})$ are estimated by $\tilde{g}_{N_1}(x^k)$ and $\tilde{g}_{N_2}(x^{k+1})$, respectively. We then reject H_0 , and hence conclude that x^{k+1} is a significant improvement over x^k , if

$$c^T x^k + \tilde{g}_{N_1}(x^k) > c^T x^{k+1} + \tilde{g}_{N_2}(x^{k+1}) + z_\alpha \left(\frac{\hat{\sigma}_{N_1}^2(x^k)}{N_1} + \frac{\hat{\sigma}_{N_2}^2(x^{k+1})}{N_2} \right)^{1/2}. \tag{33}$$

Observe that the above test requires (stochastic) independence between estimators $\tilde{g}_{N_1}(x^k)$ and $\tilde{g}_{N_2}(x^{k+1})$. However, such a condition may not always hold, as it happens in the case of the sample-path optimization discussed in Section 2. Nevertheless we can still test the above hypothesis H_0 by using a *paired t*-test as follows. Suppose that the same sample size N is used for calculation of $\tilde{g}_N(x^k)$ and $\tilde{g}_N(x^{k+1})$. We have then that the difference $\tilde{g}_N(x^k) - \tilde{g}_N(x^{k+1})$ can be represented as the average $\bar{W} := N^{-1} \sum_{i=1}^N W_i$ of appropriate independent random variables $W_i, i = 1, \dots, N$ (see (17)). Consider the sample variance $s^2 := (N - 1)^{-1} \sum_{i=1}^N (W_i - \bar{W})^2$ of W_1, \dots, W_N . Then we reject H_0 if

$$c^T (x^k - x^{k+1}) + \bar{W} > z_\alpha N^{-1/2} s. \tag{34}$$

Consider now the estimator \hat{x}_N of x_0 . Its (asymptotic) distributional properties are discussed, for example in [26,27]. The following is a particular case of more general results. Suppose that the function $f(x)$ is twice differentiable at x_0 , that the feasible set S is defined by constraints as in (24), that the constraint functions $c_i(x)$ are twice differentiable, that the gradients $\nabla c_i(x_0), i \in I(x_0)$, are linearly independent and that the strict complementarity condition holds, i.e., $\lambda_i > 0, i \in J(x_0)$. Let λ_0 and $\hat{\lambda}_N$ be the

Lagrange multipliers vectors of the true and the approximating problems, respectively. Then it is possible to show (cf. [6, p. 303]) that, under mild additional conditions,

$$N^{1/2}(\hat{x}_N - x_0, \hat{\lambda}_N - \lambda_0) \Rightarrow N(0, \Gamma), \tag{35}$$

where

$$\Gamma = \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \Psi & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix}^{-1},$$

$H = \nabla_{xx}^2 L(x_0, \lambda_0)$, with $L(x, \lambda) = f(x) + \sum_{i=1}^l \lambda_i c_i(x)$ being the Lagrangian of the true program, A is the $m \times s$ matrix whose columns are formed from the gradients $\nabla c_i(x_0), i \in I(x_0)$, and Ψ is the asymptotic covariance matrix of $N^{1/2}[\nabla \tilde{g}_N(x_0) - \nabla g(x_0)]$.

All quantities involved in the calculation of the matrix Γ can be estimated from the generated sample and hence asymptotic variances of the components of \hat{x}_N can be evaluated.

Note that, under the above assumptions, second order necessary conditions, for the ‘true’ problem at the point x_0 , can be written in the form: $x^T H x \geq 0$ for all $x \in \mathbb{R}^m$ satisfying $A^T x = 0$. The corresponding second order sufficient conditions are that $x^T H x > 0$ for all $x \neq 0$ satisfying $A^T x = 0$ (e.g. [28]). Under the above second order necessary conditions, the second order sufficient conditions are equivalent to nonsingularity of the $(m + s) \times (m + s)$ matrix

$$B = \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix}.$$

Therefore nonsingularity of B implies local uniqueness of the optimal solution x_0 . Large variances of the components of \hat{x}_N may indicate instability, or even non-uniqueness, of x_0 .

6. Implementation

In this section we study implementation aspects of the techniques discussed in the previous sections and applied to the two-stage recourse problem (3) and (4), in which only the vector h is random and has a normal distribution with mean μ and covariance matrix Σ . We describe in this section a detailed algorithm, some variance reduction techniques and present numerical results obtained for a small-sized test problem.

6.1. The algorithm

Observe initially that (3) and (4) can be rewritten as

$$\min_{x \in S, t = Tx} c^T x + E\{Q^*(t, \omega)\}, \tag{36}$$

where

$$Q^*(t, \omega) = \inf \{ q^T y : Wy = h(\omega) - t, y \geq 0 \}. \tag{37}$$

Despite the apparent increase in the dimension of the problem (which now has (x, t) as its vector of unknowns), this change-of-variables transformation has some advantages. First, the expected value function $g(\cdot) := E\{Q^*(\cdot, \omega)\}$ is strictly convex, whereas $E\{Q(\cdot, \omega)\}$ (with $Q(x, \omega)$ defined in (4)) is convex but is not strictly convex if the matrix T is singular. Consequently, for a large sample size the estimate of the Hessian matrix will indeed be positive definite, as discussed in Section 2. Furthermore, often the matrix T has more columns (which is the number of decision variables) than rows (the number of random variables). It follows that the Hessian of $g(\cdot)$ has smaller dimension than the Hessian of $E\{Q(\cdot, \omega)\}$, and typically is much better conditioned, which is important for computations.

Let $p(\cdot, \mu)$ be the normal-distribution pdf (see (18)) of h with mean μ and covariance matrix Σ (we do not write explicitly $p(\cdot, \mu)$ as a function of Σ since the covariance matrix is not changed in the process of iterations). It is clear that $p(\cdot, \mu - t)$ is the pdf of $h - t$, and hence the expected value function $g(t) := E\{Q^*(t, \omega)\}$ can be written as

$$g(t) = \int_{\mathbb{R}^n} G(y)p(y, \mu - t) dy. \tag{38}$$

As discussed in Section 2, given a so-called *reference parameter* μ_0 , we can compute the derivatives of $g(t)$ through the derivatives of $p(y, \mu - t)$. That is,

$$\begin{aligned} \nabla g(t) &= \int_{\mathbb{R}^n} \frac{G(y)}{p(y, \mu_0)} \nabla_t p(y, \mu - t)p(y, \mu_0) dy \\ &= -E_{\mu_0} \left\{ G(Y)L(Y, \mu - t, \mu_0)\Sigma^{-1}(Y - \mu + t) \right\}, \end{aligned} \tag{39}$$

and

$$\begin{aligned} \nabla^2 g(t) &= \int_{\mathbb{R}^n} \frac{G(y)}{p(y, \mu_0)} \nabla_t^2 p(y, \mu - t)p(y, \mu_0) dy \\ &= E_{\mu_0} \left\{ G(Y)L(Y, \mu - t, \mu_0) \left([\Sigma^{-1}(Y - \mu + t)] [\Sigma^{-1}(Y - \mu + t)]^T - \Sigma^{-1} \right) \right\}, \end{aligned} \tag{40}$$

where

$$L(y, \mu_1, \mu_2) = \frac{p(y, \mu_1)}{p(y, \mu_2)}$$

is the LR function. It is interesting to note that we can take $\mu_0 = \mu - t$ in (39) and (40), in which case we have $L(Y, \mu - t, \mu_0) = 1$.

Consider again the generic algorithm described in Section 3. Suppose that at the beginning of iteration k we have at hand a pair (x^k, t^k) . As discussed there, the basic idea

is to generate a random sample Y_1^k, \dots, Y_N^k , from the pdf $p(\cdot, \mu - t^k)$, and then to apply one step of a deterministic optimization algorithm to the approximating problem

$$\min_{x \in S, t = Tx} \{f_N^k(x, t) := c^T x + g_N^k(t)\}, \tag{41}$$

where

$$g_N^k(t) = N^{-1} \sum_{i=1}^N G(Y_i^k) L(Y_i^k, \mu - t, \mu - t^k). \tag{42}$$

Observe that given a sample Z_1, \dots, Z_N from the original pdf $p(\cdot, \mu)$, we can take Y_i^k to be simply equal to $Z_i - t^k$, i.e., it is not necessary to generate new random numbers. Of course, in this case Y_i^k and Y_i^{k+1} are highly correlated.

The deterministic optimization algorithm used in our implementation is a combination of successive-linear and successive-quadratic approximations for convex functions. Both methods are well known in nonlinear programming and are described in [29], for example. The idea is to minimize the linear approximation of each $f_N^k(x, t)$ until two consecutive values $f(x^k, t^k)$ and $f(x^{k+1}, t^{k+1})$ are sufficiently close to each other in the sense of the statistical paired t -test described in Section 5. From that iteration on we use a sequential quadratic programming method. It is clear that in either case we have to impose bounds on x in order to avoid a large error due to the respective approximation (linear or quadratic), in other words, it is necessary to use yet another type of trust regions for each optimization step. In [29] one can find a more detailed description of trust region methods, including a standard algorithm for updating the size of the region depending on the quality of the approximation.

Below is a detailed description of the algorithm. We adopt the notation $g_N(t^k)$ to mean the random variable defined in (42), viewing Y_1^k, \dots, Y_N^k , as i.i.d. random variables with pdf $p(\cdot, \mu - t^k)$. A realization of this random variable is denoted by $g_N^k(t^k)$. A similar notation is used for the gradient and Hessian of $g_N(t^k)$.

Algorithm

Let β and Δ^0 be pre-specified constants.

$x^0 :=$ initial guess (not necessarily feasible); $t^0 := Tx^0$;

$k := 0$;

$N :=$ small sample size;

Generate a sample Z_1, \dots, Z_N from $p(\cdot, \mu)$.

Compute $Y_i^0 := Z_i - t^0, i = 1, \dots, N$.

Compute $g_N^0(t^0), \nabla g_N^0(t^0)$ and $\nabla^2 g_N^0(t^0)$ using Z_1, \dots, Z_N .

Repeat

 Compute linear/quadratic approximation:

 Let $l^k(x, t) := c^T x + g_N^k(t^k) + \nabla g_N^k(t^k)^T (t - t^k)$.

 Let $q^k(x, t) := c^T x + g_N^k(t^k) + \nabla g_N^k(t^k)^T (t - t^k) + \frac{1}{2} (t - t^k)^T \nabla^2 g_N^k(t^k) (t - t^k)$.

 Compute size of trust region for nonlinear algorithm (only if $k \geq 1$):

 Let $R^k := [f_N^{k-1}(x^{k-1}, t^{k-1}) - f_N^k(x^k, t^k)] / [f_N^{k-1}(x^{k-1}, t^{k-1}) - q^{k-1}(x^k, t^k)]$.

 If $R^k < 0.25$ {no or insufficient decrease},

then $\Delta^k := \|t^k - t^{k-1}\|_\infty / 4$,
 else if $R^k > 0.75$ and $\|t^k - t^{k-1}\|_\infty = \Delta^{k-1}$ {binding constraint},
 then $\Delta^k := 2\Delta^{k-1}$,
 else $\Delta^k := \Delta^{k-1}$.

Compute next point:

Solve

$\min l^k(x, t)$ (or $q^k(x, t)$) subject to $x \in S, Tx = t$ and $\|t - t^k\|_\infty \leq \Delta^k$.

Let $(x^{k+1}, t^{k+1}) :=$ solution of the above problem.

$k := k + 1$;

Compute $Y_i^k := Z_i - t^k, i = 1, \dots, N$.

Compute $g_N^k(t^k), \nabla g_N^k(t^k)$ and $\nabla^2 g_N^k(t^k)$ using Y_1^k, \dots, Y_N^k .

Test need for resampling and optimality:

If $f_N^k(x^k, t^k)$ and $f_N^{k-1}(x^{k-1}, t^{k-1})$ satisfy the paired t-test for proximity (see (34)),

then Generate a new sample Z_1, \dots, Z_N from $p(\cdot, \mu)$.

Recompute $Y_i^k := Z_i - t^k, i = 1, \dots, N, g_N^k(t^k), \nabla g_N^k(t^k)$ and $\nabla^2 g_N^k(t^k)$.

Compute new sample size N' according to (22) and (23).

If $N' > N$, then

Extend sample Z_1, \dots, Z_N to $Z_1, \dots, Z_{N'}$.

Compute $Y_i^k := Z_i - t^k, i = N + 1, \dots, N'; N := N'$.

Update values of $g_N^k(t^k), \nabla g_N^k(t^k)$ and $\nabla^2 g_N^k(t^k)$.

Test optimality of x^k :

Compute $C(x^k)$ as defined in (26).

Let V^k denote the random variable $c + T^T \nabla g_N^k(t^k)$.

Let T_2 denote the random variable $\min_{z \in C(x^k)} \|V^k - z\|_2^2$.

If strict complementarity holds for x^k , then

Compute $v^k := c + T^T \nabla g_N^k(t^k), v := \min_{z \in C(x^k)} \|v^k - z\|_2^2$.

If $P(T_2 > v) \geq \beta$ {if p -value is large enough}, then **stop**.

Until {stopping criterion is satisfied}.

6.2. Variance reduction techniques

Another important topic refers to the quality of the estimators of the gradient and the Hessian of the objective function $f(x)$ at each iteration point $x = x^k$. These estimators should be accurate enough for the algorithm to proceed in significant reduction in the value of the objective function. Obviously that may not happen if the employed estimators are poor. Indeed, some numerical experiments have shown that if the variance of those estimators is too large, then the algorithm may not converge even when started at a point relatively close to the optimizer. Also, a bad estimator of the gradient can result in misleading conclusions obtained from the statistical optimality test described in Section 4. Furthermore, we are also interested in determining sharp bounds for the expected value function at each iteration in order to detect a significant reduction in the value $f(x^{k+1})$ (see Section 5) and ultimately to determine

good bounds for the optimum value. In order to overcome those difficulties we have implemented some techniques for variance reduction. Note that it suffices to study the function $g(\cdot)$ rather than $f(\cdot)$, since they differ only by a deterministic term. In what follows we describe methods used for reducing the variance of estimators of the gradient, the Hessian and the value, respectively, of the expected value function $g(\cdot)$ at the iteration points.

6.2.1. *Obtaining smaller variance estimators of the gradient $\nabla g(t)$*

The following are techniques of linear control variables (e.g. [6]). Consider the expression for $\nabla g(t)$ given in (39), and suppose that we want to estimate the gradient at $t = t^k$. For the sake of simplicity, let us assume that $\mu_0 = \mu - t^k$. Define the random vector $Z = \Sigma^{-1}(Y - \mu + t^k)$, where $Y \sim N(\mu - t^k, \Sigma)$. Then, $Z \sim N(0, \Sigma^{-1})$ and hence for any $\alpha \in \mathbb{R}$ we can write

$$V := \nabla g(t^k) = -E_{\mu-t^k} \{G(Y)Z\} = -E_{\mu-t^k} \{(G(Y) - \alpha)Z\},$$

which yields the unbiased estimator

$$\tilde{V}(\alpha) = -N^{-1} \sum_{i=1}^N (G(Y_i) - \alpha)Z_i. \tag{43}$$

The goal is to choose α in such a way that the components of $\tilde{V}(\alpha)$ have smaller variances than the components of the original estimator $\nabla g_N(t^k)$, which is equal to $\tilde{V}(\alpha)$ for $\alpha = 0$. This can be accomplished by choosing α^* that minimizes the trace of the covariance matrix $\text{Cov}(\tilde{V}(\alpha))$. Some algebraic manipulations show that

$$\alpha^* = \frac{\text{Trace} [\text{Cov}(G(Y)Z, Z)]}{\text{Trace} [\text{Cov} Z]} = \frac{E_{\mu-t^k} \{G(Y)Z^T Z\}}{\text{Trace} [\Sigma^{-1}]},$$

which can be consistently estimated by

$$\hat{\alpha}^* = \frac{N^{-1} \sum_{i=1}^N G(Y_i)Z_i^T Z_i}{\text{Trace}[\Sigma^{-1}]}. \tag{44}$$

Our numerical experience shows that these techniques are in general very effective, allowing variance reductions in the order of up to a thousand times in some cases.

Further improvement is achieved by estimating $\nabla g(t^k)$ through the subgradients of $G(h - t^k)$, as discussed in Section 2 (see (10)). Let

$$\tilde{V}^s = -N^{-1} \sum_{i=1}^N S_i,$$

where S_i is a subgradient of $G(\cdot)$ at Y_i , i.e. S_i is an optimal solution of (6) for $z = Y_i$. Then \tilde{V}^s is an unbiased estimator of $\Delta g(t^k)$ and so is

$$\tilde{W}(\lambda) = \lambda \tilde{V}^s + (1 - \lambda) \tilde{V}(\alpha^*)$$

for any $\lambda \in \mathbb{R}$. Similarly as above, we can choose λ so as to minimize the trace of the covariance matrix $\text{Cov}(\tilde{W}(\lambda))$. This optimal choice is given by

$$\lambda^* = \frac{\text{Trace} [\text{Cov}(\tilde{V} - \tilde{V}^s, \tilde{V}^s)]}{\text{Trace} [\text{Cov}(\tilde{V} - \tilde{V}^s)]} = \frac{E\{(\tilde{V} - \tilde{V}^s)^\top \tilde{V}^s\}}{E\{(\tilde{V} - \tilde{V}^s)^\top (\tilde{V} - \tilde{V}^s)\}},$$

(where $\tilde{V} = \tilde{V}(\alpha^*)$), which can be estimated from the values $\tilde{V}_i^s = S_i$ and

$$\tilde{V}_i = (G(Y_i) - \alpha^*)Z_i, \quad i = 1, \dots, N.$$

Again, our computational experience has shown that typically when x^k is far from the optimizer, the function $G(\cdot - Tx^k)$ is almost linear and therefore the subgradient estimator \tilde{V}^s is extremely accurate. As x^k gets close to the optimizer the gain in variance reduction decreases.

6.2.2. *Obtaining smaller variance estimators of $\nabla^2 g(t)$*

Techniques of linear control variates can be also used to obtain a smaller-variance estimator of $\Delta^2 g(t)$ in a similar fashion. Here we consider the expression given in (40), and define the random matrix $X = ZZ^\top$, where $Z = \Sigma^{-1}(Y - \mu + t^k)$ as before. Then, X has a central *Wishart* distribution $W_m(1, \Sigma^{-1})$ and hence, since $EX = \Sigma^{-1}$, it follows that we can write for all $\beta \in \mathbb{R}$

$$U := \nabla^2 g(t^k) = E_{\mu-t^k} \{G(Y)(X - \Sigma^{-1})\} = E_{\mu-t^k} \{(G(Y) - \beta)(X - \Sigma^{-1})\}.$$

This yields the unbiased estimator

$$\tilde{U}(\beta) = N^{-1} \sum_{i=1}^N (G(Y_i) - \beta)(X_i - \Sigma^{-1}). \tag{45}$$

As with the estimator of the gradient $V(\alpha)$, we want to minimize the sum of the variances of the components of $U(\beta)$. Clearly, this is equivalent to minimizing $\text{Trace} [\text{Cov}(\text{Vec}(\tilde{U}(\beta)))]$, where $\text{Vec}(A)$ is the *vector* operator which maps a matrix $A_{m \times n}$ into a single vector $a_{mn \times 1}$ formed by stacking columns of A . The optimal value β^* is given by

$$\begin{aligned} \beta^* &= \frac{\text{Trace}[\text{Cov}(\text{Vec}(G(Y)(X - \Sigma^{-1})), \text{Vec}(X - \Sigma^{-1}))]}{\text{Trace}[\text{Cov}(\text{Vec}(X))]} \\ &= \frac{E\{\text{Trace} [G(Y)(X - \Sigma^{-1})^2]\}}{(\text{Trace} [\Sigma^{-1}])^2 + \text{Trace} [\Sigma^{-2}]}, \end{aligned}$$

which can be estimated by using the values Y_i and $X_i, i = 1, \dots, N$, and taking the corresponding average. In our experiments the reduction obtained with this technique was in general of the same order as the reduction obtained for the gradient.

6.2.3. *Obtaining smaller variance estimators of $g(t)$*

Techniques used to obtain a better estimator of $g(t)$ for $t = t^k$ differ completely from the ones discussed above. The main reason for the dissimilarity stems from the fact that, contrary to the gradient and Hessian, the value of the function does

not play a significant role in finding descent directions, which means that there is no need for a high precision of the estimator of $g(t^k)$ when t^k is far from the optimizer. On the other hand, when t^k is close to the optimizer, it is important to determine smaller confidence intervals in order to detect significant reduction in the value $g(t^k)$ at successive iterations and also to determine good bounds for the optimum value if the current iteration t^k is accepted as an optimal solution.

With this idea in mind, suppose that the current point (x^k, t^k) is a candidate for the optimal solution in the sense that t^k is close enough to the previous point t^{k-1} (see the algorithm in Section 6.1). Consider the estimator $g_N(t^k)$, given in (42), computed with the sample Y_1, \dots, Y_N from the pdf $p(\cdot, \mu - t^k)$. Alternatively, given a parameter μ_0 , consider the estimator $\bar{g}_N(t^k, \mu_0)$ computed in the same way as $g_N(t^k)$ except that the sample is taken from the pdf $p(\cdot, \mu_0)$. As was shown in Section 2, the variance of $\bar{g}_N(t^k, \mu_0)$ is given by

$$\begin{aligned} \sigma^2(\mu_0) &= N^{-1} \left[E_{\mu_0} \{G(Y)^2 L(Y, \mu - t^k, \mu_0)^2\} - g(t)^2 \right] \\ &= N^{-1} \left[E_{\mu - t^k} \{G(Y)^2 L(Y, \mu - t^k, \mu_0)\} - g(t)^2 \right]. \end{aligned}$$

The goal is to choose μ_0 in order to minimize $\sigma^2(\mu_0)$. As discussed in [6], $\sigma^2(\cdot)$ is strictly convex and therefore has a unique minimizer μ_0^* . Note that the term $g(t)$ can be dropped from this minimization problem as it is constant with respect to μ_0 . By applying again the underlying idea of stochastic counterpart we can then minimize the approximation

$$\psi(\mu_0) = N^{-2} \sum_{i=1}^N G(Y_i)^2 L(Y_i, \mu - t^k, \mu_0).$$

In our implementation we used a sequential quadratic programming method to solve this subproblem and find μ_0^* .

We can also combine the above procedure with the linear control variate techniques as follows. Let L denote the random variable $L(Y, \mu - t^k, \mu_0^*)$, where $Y \sim N(\mu_0^*, \Sigma)$. Then, L has a lognormal distribution with mean 1 and variance $e^\delta - 1$, where

$$\delta = (\mu - t^k - \mu_0^*)^T \Sigma^{-1} (\mu - t^k - \mu_0^*).$$

Note that for any $\rho \in \mathbb{R}$ we can write

$$R := g(t^k) = E_{\mu_0^*} \{G(Y)L\} = E_{\mu_0^*} \{G(Y)L - \rho(L - 1)\},$$

so that the estimator

$$\tilde{R}(\rho) = N^{-1} \sum_{i=1}^N G(Y_i)L_i - \rho(L_i - 1) \tag{46}$$

is unbiased. Again, we can find ρ^* that minimizes the variance of $\tilde{R}(\rho)$, that is

$$\rho^* = \frac{\text{Cov}(G(Y)L, L - 1)}{\text{Var } L} = \frac{E\{G(Y)L(L - 1)\}}{e^\delta - 1}.$$

As before, ρ^* can be estimated by computing $G(Y_i)L_i(L_i - 1)$, $i = 1, \dots, N$, and taking the corresponding average. This strategy has also been implemented in our algorithm.

6.3. Numerical results

We tested the method discussed in this work on a collection of test problems provided to us by Dr. János Mayer from University of Zurich. Since the results obtained for most problems were similar (in the sense of precision), we chose one of them as a representative in order to illustrate the type of analysis that can be made and describe the results obtained.

The problem has the general format given by (3) and (4) (except that the matrix T is deterministic), with the set S given by $\{x: Ax = b, x \geq 0\}$. The data for c, A, b, W, q and T were randomly produced by a generator and are as follows:

$$c = [0.73 \quad -2.16 \quad -0.31 \quad 9.00 \quad -5.33 \quad 4.30 \quad 5.80 \quad 6.17 \quad -0.09 \quad 2.65]^T,$$

$$A = \begin{bmatrix} 0 & -4.19 & 0 & 0 & 0 & 4.12 & 0 & 0 & -3.53 & 0 \\ -0.34 & -1.88 & 0 & 0 & 0 & 0 & -1.32 & 0 & 0 & -4.54 \\ 0 & 3.04 & 8.34 & 3.41 & -7.90 & 0 & 0 & 6.45 & 0 & 9.80 \\ 0 & 0 & -9.97 & 0 & 0 & 0 & 5.26 & 0 & 0 & -0.89 \\ 0 & -0.92 & 0 & 6.57 & 0 & 0 & 2.05 & 0 & 2.17 & -2.31 \end{bmatrix},$$

$$b = [-1.24 \quad -2.79 \quad 8.00 \quad -1.94 \quad 2.61]^T,$$

$$W = \begin{bmatrix} 0 & 0 & 0 & -0.07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.17 & 5.48 & 0 & 0 & 0 & -7.35 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.93 & -7.41 & 0 & 0 & 0 \\ 0 & 4.36 & 0 & 0 & 0 & 9.69 & 0 & 5.35 & 0 & 0 \\ 0 & 0 & 0 & -1.60 & 0 & 0 & 0 & 0 & 0 & 7.43 \\ -2.30 & 0 & 0 & 5.38 & 2.96 & 0 & 0 & 0 & 0 & 3.74 \\ 0 & 5.13 & 0 & 0 & 2.15 & 1.65 & 0 & 0 & -5.73 & 2.09 \\ 0 & 0 & 0 & -2.74 & 0 & 0 & 0 & 0 & -2.58 & 0 \\ 0.18 & 0 & 0 & -5.49 & -7.52 & -8.92 & 0 & 0 & 0 & 8.96 \\ -6.37 & 0 & 0 & -3.05 & 0 & 0 & 0 & 0 & 0 & -5.88 \\ 8.49 & 8.96 & 3.45 & 0 & 2.41 & -10.36 & 0 & 2.69 & 0 & 0 \\ 0 & 0 & -6.02 & 0.29 & 0 & 0 & 7.41 & -0.69 & 5.27 & 0 \\ 0 & -18.45 & 0 & 0 & 0 & 0 & 0 & 0 & 3.05 & -16.36 \\ 0 & 0 & 0 & 1.80 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -6.60 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

$$q = [0. \quad 0. \quad 4.80 \quad 5.99 \quad 9.46 \quad 7.01 \quad 0. \quad 6.46 \quad 2.88 \quad 0. \quad 0. \quad 4.95 \quad 0. \quad 0. \quad 1.29]^T,$$

$$T = \begin{bmatrix} -8.42 & 0 & 0 & 0 & 0 & 0 & 6.91 & 0 & -2.07 & 0 \\ -5.23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2.14 & 0 & 0 & 0 & 1.16 & 0 & 0 & 0 & 0 & 0 \\ 1.19 & 0 & 0 & 0 & 0 & 0 & 0 & -6.05 & 0 & -4.82 \\ 0 & 0 & 5.48 & 0 & 0 & 0 & 0 & -4.75 & 0 & 0 \\ 2.38 & 0 & 2.90 & 0 & 0 & -0.88 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.04 & 0 & 0 \\ 2.61 & 0 & 0 & -0.91 & 0 & 0 & 0 & 0 & -4.93 & 0 \\ -5.79 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.60 & 0 \\ -2.64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The vector h has a normal distribution with mean vector

$$\mu = [-3.88 \quad 1.12 \quad -4.63 \quad 5.04 \quad 2.05 \quad 5.19 \quad -5.53 \quad 3.80 \quad 1.81 \quad -9.29]^T.$$

The components of h are independent with standard deviations

$$\sigma = [0.15 \quad 0.01 \quad 0.21 \quad 0.25 \quad 0.04 \quad 0.27 \quad 0.31 \quad 0.14 \quad 0.03 \quad 0.86]^T.$$

The initial point chosen was $x^0 = [1, 1, \dots, 1]^T$.

Table 1 shows the results obtained. At the end of each iteration k , we list the ten components of the current solution x^k , the value of the estimate $f_N^k(x^k, t^k)$, the half-width of a 95% confidence interval (Δ) for $f_N^k(x^k, t^k)$ and the value of the statistic $T_2 = \min_{z \in C(x^k)} \|\nabla f_N^k(x^k, t^k) - z\|_2^2$ with its corresponding p -value, which is an indication of the proximity of x^k to the optimal solution (see Section 4). The last column displays the sample size used for that iteration.

For reasons of space, we list only the iterations (except for the first one) where a new sample was generated, i.e., whenever the current point x^k was ‘close’ enough to the previous point x^{k-1} in the sense of the paired t -test described in Section 5. Note also that on those iterations a reference parameter optimization was performed to reduce the variance of the estimator $f_N(x^k, t^k)$ (see discussion under the topic ‘Variance reduction techniques’). Finally, a limit of 5000 was imposed on the maximum sample size used.

Observe that the p -value obtained in the last iteration indicate that the corresponding solution can be accepted as optimal (i.e., the hypothesis ‘ x^{21} is optimal’ is not rejected) with a level of significance approximately equal to 0.77, which is a strong evidence of optimality. We can then compute *individual* 95% confidence intervals for each component of the optimal solution by using the distributional result given in (35), obtaining lower and upper bounds for these components. One should note, however, that those bounds are only an *indication* of the involved stochastic error, since their calculation is based on the assumption that the deterministic error (the bias) of the current estimators is *significantly smaller* than the stochastic error measured by the corresponding variances. Table 2 displays these results.

For the sake of comparison, we also list the solution obtained by János Mayer. He provided an optimal point as well as lower and upper bounds for the optimim value. Those results are very close to what we have obtained, as Table 3 shows.

Table 1
Evolution of the algorithm

Iter.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.913392	0.000000	0.145255	0.449450	0.436054	0.064256	0.000000	0.516616
14	0.534576	0.000000	0.142719	0.464959	0.090342	0.49869	0.000000	0.045191
15	0.533611	0.000000	0.142713	0.463483	0.088718	0.053764	0.000000	0.043880
21	0.522541	0.000000	0.142639	0.462186	0.053281	0.057888	0.000000	0.000000

Iter.	x_9	x_{10}	$f_N(x^k, t^k)$	Δ	T_2 Stat.value	p -Value	N
1	0.426984	0.547548	20.910082	0.166250	147.177979	0.0000	50
14	0.410195	0.575877	15.290671	0.109019	86.200317	0.0000	390
15	0.414740	0.575949	15.305968	0.022129	20.644712	0.0000	3900
21	0.419552	0.576777	15.186682	0.017366	0.161485	0.7699	5000

Table 2
Bounds for the optimal solution found

	Lower bound	Estimate	Upper bound
x_1	0.504499	0.522541	0.540583
x_2	0.000000	0.000000	0.000000
x_3	0.142518	0.142639	0.142760
x_4	0.457614	0.462186	0.466758
x_5	0.049780	0.053281	0.056782
x_6	0.047220	0.057888	0.068556
x_7	0.000000	0.000000	0.000000
x_8	0.000000	0.000000	0.000000
x_9	0.407103	0.419552	0.432001
x_{10}	0.575428	0.576777	0.578126

Table 3
Solution obtained with János Mayer's software

	Estimate
x_1	0.518506
x_2	0.000000
x_3	0.142612
x_4	0.463797
x_5	0.054323
x_6	0.053981
x_7	0.000000
x_8	0.000000
x_9	0.414994
x_{10}	0.577079

Lower bound for the optimum value: 14.992770.

Upper bound for the optimum value: 15.682196.

7. Conclusions

Although tested for a relatively small number of problems, the obtained numerical results are quite encouraging given the level of precision obtained. In particular, it was possible to derive sharp bounds for each component of the optimal solution, which to our knowledge is not found in previous works in the literature. It also appears that the constructed statistical optimality test is a good indication of the quality of the obtained solution. Another interesting feature of the algorithm is its relative insensitivity to the number of decision variables, which makes it a promising method for large problems. Indeed, we have tested the algorithm for problems with up to 90 decision variables and 10 random variables, and 30 decision variables and 20 random variables. In most cases the results, although not as sharp as the example in Section 6, match the ones obtained with other solvers.

There are however a number of important theoretical and practical issues which can be a subject of further investigation. The conceptual idea of the SC method discussed in Section 3 requires a careful selection of a corresponding deterministic nonlinear programming algorithm. For instance, instead of the linear/quadratic approximation technique used in our implementation, one could apply some sort of bundle-type method (cf. [5]), where the objective function is approximated by piecewise linear functions, and then later use say Newton-type methods employing quadratic approximations. A careful tuning of a chosen deterministic algorithm could be problem dependent and requires a further investigation.

Another important practical issue is how to divide the computational effort between successive steps of an iterative procedure. We have provided in this work some guidelines for the update of the sample size at each iteration, but a careful study of the technique presented is so far an open question. For the stochastic approximation (SA) method a different type of analysis is given in [30]. It is possible to show [31] that in smooth cases the considered SC and SA methods converge *asymptotically* at the same rate (provided the SA method is employed with asymptotically optimal stepsizes). This may suggest an approach to asymptotic analysis of that problem.

Stopping criteria for the type of algorithms considered here are statistical in nature and are closely related to validation analysis discussed in Section 4. Validation analysis has also an independent interest. It appears that such analysis should be different in nondifferentiable cases (cf. [32]). As mentioned earlier nondifferentiable cases appear naturally in stochastic programming with recourse and queueing systems [33]. This requires further theoretical and numerical investigations.

Acknowledgements

The authors are grateful to Dr. János Mayer for providing test problems as well as solutions obtained with his software, and to two referees whose comments helped to improve the presentation of this paper.

References

- [1] Y. Ermoliev, Stochastic quasi-gradient methods and their application to systems optimization, *Stochastics* 4 (1983) 1–37.
- [2] Y. Ermoliev, R.J.B. Wets (Eds.), *Numerical Techniques for Stochastic Optimization*. Springer, Berlin, 1988.
- [3] J.L. Higle, S. Sen, Stochastic decomposition: An algorithm for two-stage linear programs with recourse, *Mathematics of Operations Research* 16 (1991) 650–669.
- [4] G. Infanger, *Planning under Uncertainty, Solving Large Scale Stochastic Linear Programs*, Boyd & Fraser Publishing Company, MA, USA, 1994.
- [5] E.L. Plambeck, B.R. Fu, S.M. Robinson, R. Suri, Sample-path optimization of convex stochastic performance functions, *Mathematical Programming, Series B* 75 (1996) 137–176.
- [6] R.Y. Rubinstein, A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, Wiley, New York, 1993.
- [7] G. Dantzig, Linear programming under uncertainty, *Management Science* 1 (1955) 197–206.
- [8] E. Beale, On minimizing a convex function subject to linear inequalities, *Journal of the Royal Statistical Society Series B* 17 (1955) 173–184.
- [9] J. Dupačová, Multistage stochastic programs: The state-of-the-art and selected bibliography, *Kybernetika* 31 (1995) 151–174.
- [10] P. Kall, S.W. Wallace, *Stochastic Programming*, Wiley, Chichester, 1994.
- [11] R. Wets, Stochastic programming: solution techniques and approximation schemes, *Mathematical Programming: The State-of-the-Art 1982*, Springer, Berlin, 1983 pp. 566–603.
- [12] R. Wets, Stochastic programs with fixed recourse: the equivalent deterministic program, *SIAM Review* 16 (1974) 309–339.
- [13] G. Gürkan, A.Y. Özge, S.M. Robinson, Sample-path optimization in simulation, *Proceedings of the 1994 Winter Simulation Conference*, 247–254.
- [14] R.Y. Rubinstein, A. Shapiro, Optimization of static simulation models by the score function method, *Mathematics and Computers in Simulation* 32 (1990) 373–392.
- [15] R.T. Rockafellar, R.J.-B. Wets, On the interchange of subdifferentiation and conditional expectation for convex functionals, *Stochastics* 7 (1982) pp. 173–182.
- [16] R.Y. Rubinstein, Sensitivity analysis of discrete event systems by the push-out method, *Annals of Operations Research* 39 (1992) 229–250.
- [17] R.Y. Rubinstein, A. Shapiro, On optimal choice of reference parameters in the likelihood method, *Proceedings of the 1992 Winter Simulation Conference*, 1992, pp. 515–520.
- [18] W. Römisch, R. Schultz, Stability of solutions for stochastic programs with complete recourse. *Mathematics of Operations Research* 18 (1993) 590–609.
- [19] A. Shapiro, Y. Wardi, ‘Convergence analysis of stochastic algorithms’, *Mathematics of Operations Research* 21 (1996) 615–628.
- [20] R.J. Muirhead, *Aspects of Multivariate Statistical Theory*, Wiley, New York, 1982.
- [21] J.L. Higle, S. Sen, Statistical verification of optimality conditions for stochastic programs with recourse, *Annals of Operations Research* 30 (1991) 215–240.
- [22] A.M. Mathai, S.B. Provost, *Quadratic Forms in Random Variables: Theory and Applications*, Dekker, New York, 1992.
- [23] T. Robertson, F.T. Wright, R.L. Dykstra, *Order restricted Statistical Inference*, Wiley, New York, 1988.
- [24] A. Shapiro, Towards a unified theory of inequality constrained testing in multivariate analysis, *International Statistical Review* 56 (1988) 49–62.
- [25] A. Shapiro, Asymptotic analysis of stochastic programs, *Annals of Operations Research* 30 (1991) 169–186.
- [26] A.J. King, R.T. Rockafellar, Asymptotic theory for solutions in statistical estimation and stochastic programming, *Mathematics of Operations Research* 18 (1993) 148–162.
- [27] A. Shapiro, Asymptotic behavior of optimal solutions in stochastic programming, *Mathematics of Operations Research* 18 (1993) 829–845.
- [28] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

- [29] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, 1993.
- [30] P.L'Ecuyer, G. Yin, Budget-dependent convergence rate of stochastic approximation, Preprint.
- [31] A. Shapiro, Simulation based optimization – convergence analysis and statistical inference, *Stochastic Models* 12 (1996) 425–454.
- [32] J.L. Hige, S. Sen, Duality and statistical tests of optimality for two stage stochastic programs, *Mathematical Programming, Series B* 75 (1996) 257–275.
- [33] A. Shapiro, Y. Wardi, Nondifferentiability of the steady-state function in Discrete Event Dynamic Systems, *IEEE Transactions on Automatic Control* 39 (1994) 1707–1711.