Proceedings of the International Congress of Mathematicians
Hyderabad, India, 2010

Computational Complexity of Stochastic Programming: Monte Carlo Sampling Approach

Alexander Shapiro*

Abstract

For a long time modeling approaches to stochastic programming were dominated by scenario generation methods. Consequently the main computational effort went into development of decomposition type algorithms for solving constructed large scale (linear) optimization problems. A different point of view emerged recently where computational complexity of stochastic programming problems was investigated from the point of view of randomization methods based on Monte Carlo sampling techniques. In that approach the number of scenarios is irrelevant and can be infinite. On the other hand, from that point of view there is a principle difference between computational complexity of two and multistage stochastic programming problems – certain classes of two stage stochastic programming problems can be solved with a reasonable accuracy and reasonable computational effort, while (even linear) multistage stochastic programming problems seem to be computationally intractable in general.

Mathematics Subject Classification (2010). Primary 90C15; Secondary 90C60.

Keywords. Stochastic programming, Monte Carlo sampling, sample average approximation, dynamic programming, asymptotics, computational complexity, stochastic approximation.

1. Introduction

Origins of Stochastic Programming are going back to more than 50 years ago to papers of Beale [2] and Dantzig [4]. The essential idea of that approach is

*This research was partly supported by the NSF award DMS-0914785 and ONR award N000140811104.

Georgia Institute of Technology, Atlanta, Georgia 30332, USA.
E-mail: ashapiro@isye.gatech.edu.

that decision variables are divided into groups of “here-and-now” decision variables which should be made before a realization of the uncertain data becomes available, and “wait-and-see” decision variables made after observing data and which are functions of the data. Furthermore, the uncertain parameters are modeled as random variables, with a specified probability distribution, and consequently the optimization problem is formulated in terms of minimizing the expected values of the uncertain costs.

Two-stage stochastic linear programming problems can be written in the form

$$\operatorname*{Min}_{x\in\mathcal{X}} \langle c, x\rangle + \mathbb{E}[Q(x,\xi)], \tag{1.1}$$

where $\mathcal{X} = \{x \in \mathbb{R}^n : Ax \le b\}$ and $Q(x,\xi)$ is the optimal value of the second stage problem

$$\operatorname*{Min}_{y\in\mathbb{R}^m} \langle q, y\rangle \ \text{ subject to } Tx + Wy \le h. \tag{1.2}$$

Some/all of the parameters, summarized in data vector $\xi := (q, h, T, W)$, of the second stage problem (1.2) are unknown (uncertain) at the first stage when a “here-and-now” decision x should be made, while second stage decisions $y = y(\xi)$ are made after observing the data and are functions of the data parameters. Parameters of the second stage problem are modeled as random variables and the expectation in (1.1) is taken with respect to a specified distribution of the random vector ξ.

This can be extended to the following multistage setting of T-stage stochastic programming problems

$$\operatorname*{Min}_{x_1\in\mathcal{X}_1} f_1(x_1) + \mathbb{E}\left[\inf_{x_2\in\mathcal{X}_2(x_1,\xi_2)} f_2(x_2,\xi_2) + \mathbb{E}\Big[\cdots + \mathbb{E}\Big[\inf_{x_T\in\mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T,\xi_T)\Big]\Big]\right], \tag{1.3}$$

driven by the random data process $\xi_1, \xi_2, \ldots, \xi_T$. Here $x_t \in \mathbb{R}^{n_t}$, $t = 1, \ldots, T$, are decision variables, $f_t : \mathbb{R}^{n_t} \times \mathbb{R}^{d_t} \to \mathbb{R}$ are continuous functions and $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{d_t} \rightrightarrows \mathbb{R}^{n_t}$, $t = 2, \ldots, T$, are measurable closed valued multifunctions. The first stage data, i.e., the vector ξ_1, the function $f_1 : \mathbb{R}^{n_1} \to \mathbb{R}$, and the set $\mathcal{X}_1 \subset \mathbb{R}^{n_1}$ are deterministic (not random). It is said that the multistage problem is *linear* if the objective functions and the constraint functions are linear. That is,

$$f_t(x_t,\xi_t) := \langle c_t, x_t\rangle, \ \ \mathcal{X}_1 := \{x_1 : A_1x_1 \le b_1\}, \tag{1.4}$$

$$\mathcal{X}_t(x_{t-1},\xi_t) := \{x_t : B_tx_{t-1} + A_tx_t \le b_t\},\ t = 2, \ldots, T, \tag{1.5}$$

where $\xi_1 := (c_1, A_1, b_1)$ is known at the first stage (and hence is nonrandom), and $\xi_t := (c_t, B_t, A_t, b_t) \in \mathbb{R}^{d_t}$, $t = 2, \ldots, T$, are data vectors.

For a long time approaches to modeling and solving stochastic programming problems were dominated by scenario generation methods. In such an approach a finite number of scenarios, representing what may happen in the future with assigned probability weights, were generated and consequently the constructed optimization problem was solved by decomposition type methods. If one takes the position that generated scenarios represent reality in a reasonably accurate way, then there is no dramatic difference between two and multistage stochastic programming. An argument is that considering many scenarios is certainly better than solving the problem for just one scenario which would be a deterministic optimization approach. Everybody would agree, however, that what will really happen in the future will be different with probability one (w.p.1) from the set of generated scenarios. This raises the question of what does it mean to solve a stochastic programming problem? In that respect we may cite [3, p. 413]: "... it is absolutely unclear what the resulting solution [of a scenario based approximation of a multistage stochastic program] has to do with the problem we intend to solve. Strictly speaking , we even cannot treat this solution as a candidate solution, bad or good alike, to the original problem – the decision rules we end up with simply do not say what our decisions should be when the actual realizations of the uncertain data differ from the scenario realizations."

Somewhat different point of view emerged in a number of recent publications. It was shown theoretically and demonstrated in various numerical studies that certain classes of two stage stochastic programming problems can be solved with reasonable accuracy and reasonable computational effort by employing Monte Carlo sampling techniques. From that point of view the number of scenarios is irrelevant and can be astronomically large or even infinite. On the other hand, it turns out that computational complexity of multistage stochastic programming problems is conceptually different and scenario generation methods typically fail to solve multistage stochastic problems in a reasonable sense to a "true" optimality. It also could be pointed out the criticism of the modeling assumption of knowing the "true" probability distribution of the involved random data. We will not discuss this aspect of the stochastic programming approach here.

We will use the following notation and terminology through the paper. Notation " $:=$ " means "equal by definition"; by $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$ we denote the n-dimensional simplex; $\mathbb{S}^m$ denotes the linear space of $m \times m$ symmetric matrices; $\langle x, y\rangle$ denotes the standard scalar product of two vectors $x, y \in \mathbb{R}^n$ and $\langle x, y\rangle := \mathrm{Tr}(xy)$ for $x, y \in \mathbb{S}^m$; unless stated otherwise $\|x\| = \sqrt{\langle x, x\rangle}$ denotes the Euclidean norm of vector x; $C^* = \{y : \langle y, x\rangle \geq 0,\ \forall x \in C\}$ denotes the (positive) dual of cone $C \subset \mathbb{R}^n$; by " $\preceq_C$ " we denote partial order induced by a closed convex cone C in a finite dimensional vector space, i.e., $x \preceq_C y$ means that $y - x \in C$; $\mathrm{int}(C)$ denotes the interior of set $C \subset \mathbb{R}^n$; $\mathrm{dist}(x, C) := \inf_{y \in C} \|x - y\|$ denotes the distance from point $x \in \mathbb{R}^n$ to set C; $\mathsf{Prob}(A)$ denotes probability of event A; $\Delta(\xi)$ denotes measure of mass one at point ξ; " $\xrightarrow{\mathcal{D}}$ " denotes convergence in distribution; $\mathcal{N}(\mu, \sigma^2)$ denotes normal distribution with mean μ and variance σ^2; $M_Y(t) := \mathbb{E}[\exp(tY)]$ is the moment generating function of random variable Y; $\mathbb{E}[X|Y]$ denotes conditional expectation of random variable X given Y, and $\mathrm{Var}[X]$ denotes variance of X.

2. Asymptotic Analysis

Consider the following stochastic optimization problem

$$\operatorname*{Min}_{x\in\mathcal{X}} \big\{ f(x) := \mathbb{E}[F(x,\xi)] \big\}. \tag{2.1}$$

Here $\mathcal{X}$ is a nonempty closed subset of $\mathbb{R}^n$, ξ is a random vector whose probability distribution P is supported on a set $\Xi \subset \mathbb{R}^d$, and $F : \mathcal{X} \times \Xi \to \mathbb{R}$. Unless stated otherwise all probabilistic statements will be made with respect to the distribution P. The two stage problem (1.1) is of that form with $F(x,\xi) := \langle c, x\rangle + Q(x,\xi)$. We assume that the expectation $f(x)$ is well defined and finite valued for every $x \in \mathbb{R}^n$. This, of course, implies that $F(x,\xi)$ is finite valued for almost every (a.e.) $\xi \in \Xi$. For the two stage problem (1.1) the later means that the second stage problem (1.2) is bounded from below and its feasible set is nonempty for a.e. realization of the random data.

Suppose that we have a sample $\xi^1, ..., \xi^N$ of N realizations of the random vector ξ. We assume that the sample is iid (independent identically distributed). By replacing the "true" distribution P with its empirical estimate $P_N := \frac{1}{N}\sum_{j=1}^N \Delta(\xi^j)$, we obtain the following approximation of the "true" problem (2.1):

$$\operatorname*{Min}_{x\in\mathcal{X}} \left\{ \hat{f}_N(x) := \tfrac{1}{N}\sum_{j=1}^N F(x,\xi^j) \right\}. \tag{2.2}$$

We denote by ϑ^* and $\hat{\vartheta}_N$ the optimal values of problems (2.1) and (2.2), respectively, and by $\mathcal{S}$ and $\mathcal{S}_N$ the respective sets of optimal solutions.

In the recent literature on stochastic programming, problem (2.2) is often referred to as the Sample Average Approximation (SAA) problem, and in machine learning as the empirical mean optimization. The sample $\xi^1, ..., \xi^N$ can be a result of two somewhat different procedures – it can be given by a historical data of observations of ξ, or it can be generated in the computer by Monte Carlo sampling techniques. We will be mainly interested here in the second case where we view the SAA problem (2.2) as an approach to solving the true problem (2.1) by randomization techniques.

By the Law of Large Numbers (LLN) we have that for any $x \in \mathcal{X}$, $\hat{f}_N(x)$ tends to $f(x)$ w.p.1 as $N \to \infty$. Moreover, let us assume the following.

(A1) For any $x \in \mathcal{X}$ the function $F(\cdot,\xi)$ is continuous at x for a.e. $\xi \in \Xi$.

(A2) There exists an integrable function $H(\xi)$ such that $|F(x,\xi)| \le H(\xi)$ for all $x \in \mathcal{X}$ and $\xi \in \Xi$.

These assumptions imply that $f(x)$ is continuous on $\mathcal{X}$ and $\hat{f}_N(x)$ converges w.p.1 to $f(x)$ uniformly on any compact subset of $\mathcal{X}$ (uniform LLN). Assuming further that $\mathcal{X}$ is compact, it is not difficult to show that the optimal value $\hat{\vartheta}_N$ and an optimal solution $\hat{x}_N$ of the SAA problem converge to their true counterparts w.p.1 as $N \to \infty$ (see, e.g., [20, section 5.1.1.]).

It is also possible to derive rates of convergence. Let us make the following stronger assumptions.

(A3) For some point $x^* \in \mathcal{X}$ the expectation $\mathbb{E}[F(x^*, \xi)^2]$ is finite.

(A4) There exists a measurable function $C(\xi)$ such that $\mathbb{E}[C(\xi)^2]$ is finite and

$$|F(x,\xi) - F(x',\xi)| \le C(\xi)\|x - x'\|, \ \ \forall x, x' \in \mathcal{X}, \ \forall \xi \in \Xi.$$

Suppose further that the set $\mathcal{X}$ is compact and consider Banach space $C(\mathcal{X})$ of continuous functions $\phi : \mathcal{X} \to \mathbb{R}$. Then $\hat{f}_N$ can be viewed as a random element of $C(\mathcal{X})$, and $N^{1/2}(\hat{f}_N - f)$ converges in distribution to a random element $Y \in C(\mathcal{X})$. This is the so-called functional Central Limit Theorem (CLT) (e.g., [1]). By employing further an infinite dimensional Delta Theorem it is possible to derive the following result (cf., [17]).

Theorem 2.1. *Suppose that the set $\mathcal{X}$ is compact and assumptions* (A3) *and* (A4) *hold. Then $N^{1/2}(\hat{f}_N - f)$ converges in distribution to a random element $Y \in C(\mathcal{X})$ and*

$$\hat{\vartheta}_N = \inf_{x \in \mathcal{S}} \hat{f}_N(x) + o_p(N^{-1/2}), \tag{2.3}$$

$$N^{1/2}(\hat{\vartheta}_N - \vartheta^*) \xrightarrow{\mathcal{D}} \inf_{x \in \mathcal{S}} Y(x). \tag{2.4}$$

If, moreover, $\mathcal{S} = \{\bar{x}\}$ is a singleton, then

$$N^{1/2}(\hat{\vartheta}_N - \vartheta^*) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma^2), \tag{2.5}$$

where $\sigma^2 := \mathrm{Var}[F(\bar{x}, \xi)]$.

The above result shows that the optimal value of the SAA problem converges to the optimal value of the true problem at a stochastic rate of $O_p(N^{-1/2})$. In particular, if the true problem has unique optimal solution $\bar{x}$, then $\hat{\vartheta}_N = \hat{f}_N(\bar{x}) + o_p(N^{-1/2})$, i.e., $\hat{\vartheta}_N$ converges to ϑ^* at the same asymptotic rate as $\hat{f}_N(\bar{x})$ converges to $f(\bar{x})$.

It is not difficult to show that $\mathbb{E}[\hat{\vartheta}_N] \le \vartheta^*$ and $\mathbb{E}[\hat{\vartheta}_{N+1}] \ge \mathbb{E}[\hat{\vartheta}_N]$ (cf., [10]), i.e., $\hat{\vartheta}_N$ is a biased down estimate of ϑ^* and the bias is monotonically deceasing with increase of the sample size N. Note that for any *fixed* $x \in \mathcal{X}$ we have that $\mathbb{E}[\hat{f}_N(x)] = f(x)$ and hence $\mathbb{E}[Y(x)] = 0$, where $Y(x)$ is the random function specified in Theorem 2.1. Therefore if $\mathcal{S} = \{\bar{x}\}$ is a singleton, then the asymptotic bias of $\hat{\vartheta}_N$ is of order $o(N^{-1/2})$. On the other hand, if the true problem has more than one optimal solution, then the expected value of $\inf_{x \in \mathcal{S}} Y(x)$ typically will be strictly negative and hence the asymptotic bias will be of order $O(N^{-1/2})$.

In some situations the feasible set of stochastic program is also given in a form of expected value constraints. That is, consider the following problem

$$\operatorname*{Min}_{x\in\mathcal{X}} \big\{ f(x) := \mathbb{E}[F(x,\xi)] \big\} \text{ subject to } g(x) \preceq_C 0, \tag{2.6}$$

where $C \subset \mathbb{R}^m$ is a closed convex cone and $g(x) := \mathbb{E}[G(x,\xi)]$ with $G : \mathcal{X}\times\Xi \to \mathbb{R}^m$. Note that constraint $g(x) \preceq_C 0$ means that $-g(x) \in C$. We assume that for every $x \in \mathbb{R}^n$ the expectation $g(x)$ is well defined and finite valued. Here in addition to the data of problem (2.1) we have constraints $g(x) \preceq_C 0$. For example if $C := \mathbb{R}^m_+$, then these constraints become $g_i(x) \le 0$, $i = 1, ..., m$, where $g_i(x)$ is the i-th component of the mapping $g(x)$. If $C := \mathbb{S}^m_+$ is the cone of $m \times m$ positive semidefinite matrices and $G(x,\xi)$ is an affine in x mapping, then these constraints become constraints of semidefinite programming. Given random sample $\xi^1, ..., \xi^N$, the expected value mapping $g(x)$ can be approximated by the sample average $\hat g_N(x) := \frac{1}{N}\sum_{j=1}^N G(x,\xi^j)$, and hence the following SAA problem can be constructed

$$\operatorname*{Min}_{x\in\mathcal{X}} \hat f_N(x) \text{ subject to } \hat g_N(x) \preceq_C 0. \tag{2.7}$$

We say that problem (2.6) is convex if the set $\mathcal{X}$ is convex, and for every $\xi \in \Xi$ the function $F(\cdot,\xi)$ is convex and the mapping $G(\cdot,\xi)$ is convex with respect to the cone C, i.e.,

$$G(tx + (1-t)y, \xi) \preceq_C tG(x,\xi) + (1-t)G(y,\xi), \ \ \forall x, y \in \mathbb{R}^n, \ \forall t \in [0,1]. \tag{2.8}$$

Note that the above condition (2.8) is equivalent to the condition that $\langle \lambda, G(x,\xi)\rangle$ is convex in x for every $\lambda \in C^*$. Note also that convexity of $F(\cdot,\xi)$ and $G(\cdot,\xi)$ imply convexity of the respective expected value functions.

Consider the Lagrangian $L(x,\lambda,\xi) := F(x,\xi) + \langle\lambda, G(x,\xi)\rangle$, and its expectation $\ell(x,\lambda) := \mathbb{E}[L(x,\lambda,\xi)]$ and sample average $\hat\ell_N(x,\lambda) := \hat f_N(x) + \langle \lambda, \hat g_N(x)\rangle$, associated with problem (2.6). The Lagrangian dual of problem (2.6) is the problem

$$\operatorname*{Max}_{\lambda\in C^*} \left\{ \psi(\lambda) := \min_{x\in\mathcal{X}} \ell(x,\lambda) \right\}. \tag{2.9}$$

It is said that the Slater condition for problem (2.6) holds if there exists a point $x^* \in \mathcal{X}$ such that $g(x^*) \prec_C 0$, i.e., $-g(x^*) \in \operatorname{int}(C)$. If the problem is convex and the Slater condition holds, then the optimal values of problems (2.6) and (2.9) are equal to each other and the dual problem (2.9) has a nonempty and bounded set of optimal solutions, denoted Λ.

We can now formulate an analogue of the asymptotic result of Theorem 2.1 for convex problems of the form (2.6) (cf., [20, section 5.1.4]). We will need the following analogues of assumptions (A3) and (A4).

(A5) For some point $x^* \in \mathcal{X}$ the expectation $\mathbb{E}\left[\|G(x^*,\xi)\|^2\right]$ is finite.

(A6) There exists a measurable function $C(\xi)$ such that $\mathbb{E}[C(\xi)^2]$ is finite and

$$\|G(x,\xi) - G(x',\xi)\| \le C(\xi)\|x - x'\|, \ \ \forall x, x' \in \mathcal{X}, \ \forall \xi \in \Xi.$$

As before we denote by ϑ^* and $\hat{\vartheta}_N$ the optimal values of the true and SAA problems (problems (2.6) and (2.7)), respectively.

Theorem 2.2. *Suppose that: problem* (2.6) *is convex, Slater condition holds, the set $\mathcal{X}$ is compact and assumptions* (A3) – (A6) *are satisfied. Then*

$$\hat{\vartheta}_N = \inf_{x \in \mathcal{S}} \sup_{\lambda \in \Lambda} \hat{\ell}_N(x,\lambda) + o_p(N^{-1/2}). \tag{2.10}$$

If, moreover, $\mathcal{S} = \{\bar{x}\}$ and $\Lambda = \{\bar{\lambda}\}$ are singletons, then

$$N^{1/2}(\hat{\vartheta}_N - \vartheta^*) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\sigma^2), \tag{2.11}$$

where $\sigma^2 := \mathrm{Var}[L(\bar{x},\bar{\lambda},\xi)]$.

There is an interesting consequence of the above result. It was assumed that in the SAA problem (2.7) the *same* sample $\xi^1,...,\xi^N$ was used in constructing approximations $\hat{f}_N(x)$ and $\hat{g}_N(x)$ of the objective and constraints functions, and the asymptotic result (2.11) is formulated for that case. That is, the asymptotic variance σ^2 is given by $\mathrm{Var}[L(\bar{x},\bar{\lambda},\xi)] = \mathrm{Var}\left[F(\bar{x},\xi) + \sum_{i=1}^m \bar{\lambda}_i G_i(\bar{x},\xi)\right]$. In the Monte Carlo sampling approach we have a choice of estimating the objective function and each component of the constraint mapping $g(x)$ by using *independently* generated samples. In that case similar result holds but with the asymptotic variance given by $\mathrm{Var}\left[F(\bar{x},\xi)\right] + \sum_{i=1}^m \mathrm{Var}\left[\bar{\lambda}_i G_i(\bar{x},\xi)\right]$. Since it could be expected that the components $G_i(\bar{x},\xi)$, $i = 1,...,m$, of the constraint mapping are positively correlated with each other, in order to reduce variability of the SAA estimates it would be advantageous to use the independent samples strategy.

3. Multistage Problems

The above analysis is performed for stochastic programs of a static form (2.1) and can be applied to two stage programming problems. What can be said in that respect for dynamic programs formulated in a multistage form? A solution of the multistage program (1.3) is a policy $\bar{x}_t = \bar{x}_t(\xi_{[t]})$, $t = 1,...,T$, given by measurable functions of the data process $\xi_{[t]} := (\xi_1,...,\xi_t)$ available at the decision time $t = 2,...,T$, with $\bar{x}_1$ being deterministic. It is said that policy is feasible if it satisfies the feasibility constraints for a.e. realization of the data process, i.e., $\bar{x}_1 \in \mathcal{X}_1$ and $\bar{x}_t \in \mathcal{X}_t(\bar{x}_{t-1},\xi_t)$, $t = 2,...,T$, w.p.1.

The following dynamic programming equations can be written for the multistage program (1.3) going backward in time

$$Q_t(x_{t-1},\xi_{[t]}) = \inf_{x_t \in \mathcal{X}_t(x_{t-1},\xi_t)} \left\{ f_t(x_t,\xi_t) + \mathcal{Q}_{t+1}(x_t,\xi_{[t]}) \right\}, \ \ t = T,...,2, \tag{3.1}$$

where $\mathcal{Q}_{T+1}(x_T, \xi_{[T]}) \equiv 0$ by definition and

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) := \mathbb{E}\left\{Q_{t+1}(x_t, \xi_{[t+1]})\middle|\xi_{[t]}\right\}, \quad t = T-1, ..., 2, \tag{3.2}$$

are the respective cost-to-go functions. Finally at the first stage the following problem should be solved

$$\operatorname*{Min}_{x_1 \in \mathcal{X}_1} f_1(x_1) + \mathbb{E}[Q_2(x_1, \xi_2)]. \tag{3.3}$$

A policy $\bar{x}_t = \bar{x}_t(\xi_{[t]})$, $t = 1, ..., T$, is optimal if w.p.1 it holds that

$$\bar{x}_t \in \arg \min_{x_t \in \mathcal{X}_t(\bar{x}_{t-1}, \xi_t)} \left\{f_t(x_t, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})\right\}, \quad t = T, ..., 2, \tag{3.4}$$

and $\bar{x}_1$ is an optimal solution of the first stage problem (3.3).

Problem (3.3) looks similar to the stochastic program (2.1). The difference, however, is that for $T \geq 3$ the function $Q_2(x_1, \xi_2)$ is not given explicitly, or at least in a computationally accessible form, but in itself is a solution of a multistage stochastic programming problem. Therefore in order to solve (1.3) numerically one would need to approximate the data process $\xi_1, ..., \xi_T$ by generating a tree of scenarios. The Monte Carlo sampling approach can be employed in the following way. First, a random sample $\xi_2^1, ..., \xi_2^{N_1}$ of N_1 realizations of the random vector ξ_2 is generated. For each ξ_2^j, $j = 1, ..., N_1$, a random sample of size N_2 of ξ_3, according to the distribution of ξ_3 conditional on $\xi_2 = \xi_2^j$, is generated and so forth for later stages. We refer to this procedure as *conditional sampling*. In that way the true distribution of the random data process is discretized with every generated path of the process taken with equal probability. We refer to each generated path as scenario and to the collection of all scenarios as scenario tree. Note that the total number of scenarios $N = \prod_{t=1}^{T-1} N_t$. We denote $\mathcal{N} := \{N_1, ..., N_{T-1}\}$ and by ϑ^* and $\hat{\vartheta}_{\mathcal{N}}$ the optimal values of the true problem (1.3) and the constructed SAA problem, respectively.

Assume for the sake of simplicity that the data process is *stagewise independent*, i.e., random vector ξ_{t+1} is distributed independently of $\xi_{[t]}$, $t = 1, ..., T-1$. Then the cost-to-go functions $\mathcal{Q}_{t+1}(x_t)$, $t = 1, ..., T-1$, do not depend on the random data process. Also in that case there are two possible approaches to conditional sampling, namely for each ξ_2^j, $j = 1, ..., N_1$, it is possible to generate different samples of ξ_3 independent of each other, or it is possible to use the same sample $\xi_3^1, ..., \xi_3^{N_2}$, and so forth for later stages. We consider the second approach, which preserves the stagewise independence in the generated scenario tree, with respective samples $\xi_t^1, ..., \xi_t^{N_{t-1}}$, at stages $t = 2, ..., T$.

We can write dynamic programming equations for the constructed SAA problem. Eventually the (true) first stage problem (3.3) will be approximated by the following SAA problem

$$\operatorname*{Min}_{x_1 \in \mathcal{X}_1} f_1(x_1) + \hat{Q}_{2,N_1}(x_1), \tag{3.5}$$

where $\hat{Q}_{2,N_1}(x_1) = \frac{1}{N_1}\sum_{j=1}^{N_1}\tilde{Q}_2(x_1,\xi_2^j,\tilde{\xi})$. Here $\tilde{\xi} = (\xi_3^1,...,\xi_3^{N_2},...,\xi_T^1,...,\xi_T^{N_{T-1}})$ is random vector composed from the samples at stages $t \geq 3$ and $\tilde{Q}_2(x_1,\xi_2,\tilde{\xi})$ is the corresponding cost-to-go function of the SAA problem. Note that function $\tilde{Q}_2(x_1,\xi_2,\tilde{\xi})$ depends on the random samples used at stages $t = 3,...,T$, as well.

Suppose now that the sample size N_1 tends to infinity while sample sizes N_t, $t = 2,...,T-1$, are fixed. Then by the LLN we have that $\hat{Q}_{2,N_1}(x_1)$ converges (pointwise) w.p.1 to the function $\mathfrak{E}_2(x_1,\tilde{\xi}) := \mathbb{E}\big[\tilde{Q}_2(x_1,\xi_2,\tilde{\xi})\big|\tilde{\xi}\big]$. Consider the problem

$$\operatorname*{Min}_{x_1\in\mathcal{X}_1} f_1(x_1) + \mathfrak{E}_2(x_1,\tilde{\xi}). \tag{3.6}$$

Conditional on $\tilde{\xi}$ we can view problem (3.5) as the SAA problem associated with the (static) expected value problem (3.6). Consequently asymptotic results of section 2 can be applied to the pair of problems (3.5) and (3.6).

Denote by $\vartheta^*(\tilde{\xi})$ the optimal value of problem (3.6), and recall that $\hat{\vartheta}_{\mathcal{N}}$ denotes the optimal value of problem (3.5). We have that conditional on $\tilde{\xi}$, the SAA optimal value $\hat{\vartheta}_{\mathcal{N}}$ is a biased down estimate of $\vartheta^*(\tilde{\xi})$. Since $\mathfrak{E}_2(x_1,\tilde{\xi})$ is an SAA estimate of $\mathcal{Q}_2(x_1)$, we also have that $\mathbb{E}\big[\mathfrak{E}_2(x_1,\tilde{\xi})\big] \leq \mathcal{Q}_2(x_1)$ for every $x_1 \in \mathcal{X}_1$. It follows that $\mathbb{E}[\vartheta^*(\tilde{\xi})] \leq \vartheta^*$. Consequently the bias of the SAA estimate $\hat{\vartheta}_{\mathcal{N}}$, of the optimal value ϑ^* of the true multistage problem (1.3), will increase with increase of the number of stages. It is possible to show that for some models this bias growth exponentially with increase of the number T of stages (cf., [20, p.225]).

In order for the SAA problems to converge to the true problem all samples should be increased, i.e., all sample sizes N_t should tend to infinity. In the next section we will discuss estimates of sample sizes required to solve the true problem with a given accuracy.

4. Estimates of Stochastic Complexity

In order to solve a stochastic optimization problem of the form (2.1) one needs to evaluate expectations $\mathbb{E}[F(x,\xi)]$, given by multidimensional integrals. This, in turn, requires a discretization of (continuous) distribution of the random vector ξ. Suppose that the components of ξ are distributed independently of each other and that r points are used for discretization of the marginal distribution of each component. Then the total number of discretization points (scenarios) is r^d. That is, while the input data is proportional to rd and grows linearly with increase of the number d of random parameters, the number of scenarios increases exponentially. This indicates that deterministic optimization algorithms cannot cope with such stochastic optimization problems. And, indeed, it is shown in [5] that, under the assumption that the stochastic parameters are independently distributed, two-stage linear stochastic programming problems are $\sharp$P-hard.

The Monte Carlo sampling approach of approximating the true problem (2.1) by the corresponding SAA problem (2.2) suggests a randomization approach to solving stochastic optimization problems. In a sense the sample size N, required to solve the true problem with a given accuracy, gives an estimate of computational complexity of the considered problem. Note that the SAA approach is not an algorithm, one still needs to solve the constructed SAA problem. Numerical experiments indicate that for various classes of problems, e.g., two stage linear stochastic programs, computational effort in solving SAA problems by efficient algorithms is more or less proportional to the sample size N. Theorem 2.1 suggests that the convergence of SAA estimates is rather slow. However, the convergence does not depend directly on dimension d of the random data vector, but rather on variability of the objective function.

We proceed now to estimation of the sample size required to solve the true problem with a given accuracy $\varepsilon > 0$. Recall that it is assumed that the expectation $f(x)$ is well defined and finite valued for all $x \in \mathcal{X}$. It is said that a point $\bar{x} \in \mathcal{X}$ is an ε-optimal solution of problem (2.1) if $f(\bar{x}) \le \inf_{x \in \mathcal{X}} f(x) + \varepsilon$. We denote by $\mathcal{S}^\varepsilon$ and $\hat{\mathcal{S}}_N^\varepsilon$ the sets of ε-optimal solutions of the true and SAA problems (2.1) and (2.2), respectively. Let us make the following assumptions.

(C1) There exist constants $\sigma > 0$ and $\tau > 0$ such that

$$M_{x,x'}(t) \le \exp(\sigma^2 t^2/2), \ \ \forall t \in [-\tau, \tau], \ \forall x, x' \in \mathcal{X}, \tag{4.1}$$

where $M_{x,x'}(t)$ is the moment generating function of the random variable $[F(x', \xi) - f(x')] - [F(x, \xi) - f(x)]$.

(C2) There exists a measurable function $\kappa : \Xi \to \mathbb{R}_+$ such that its moment generating function $M_\kappa(t)$ is finite valued for all t in a neighborhood of zero and

$$|F(x, \xi) - F(x', \xi)| \le \kappa(\xi) \|x - x'\|, \ \ \forall x, x' \in \mathcal{X}, \ \forall \xi \in \Xi. \tag{4.2}$$

By Cramér's Large Deviations Theorem it follows from assumption (C2) that for any $L > \mathbb{E}[\kappa(\xi)]$ there is a positive constant $\beta = \beta(L)$ such that

$$\mathsf{Prob}(\hat{\kappa}_N > L) \le \exp(-N\beta), \tag{4.3}$$

where $\hat{\kappa}_N := N^{-1} \sum_{j=1}^N \kappa(\xi^j)$. The following estimate of the sample size is obtained by applying (pointwise) upper bound of Cramér's Large Deviations Theorem and constructing a ν-net in $\mathcal{X}$ with number of points less than $(\varrho D/\nu)^n$, where $D := \sup_{x,x' \in \mathcal{X}} \|x' - x\|$ is the diameter of the set $\mathcal{X}$ and $\varrho > 0$ is an appropriate constant (cf., [18],[20, section 5.3.2]).

Theorem 4.1. *Suppose that the set $\mathcal{X}$ has a finite diameter D and assumptions* (C1) – (C2) *hold with respective constants σ and τ, and let $\alpha \in (0, 1)$, $L >$*

$\mathbb{E}[\kappa(\xi)]$, $\beta = \beta(L)$ and $\varepsilon > 0$, $\delta > 0$ be constants such that $\varepsilon > \delta \geq 0$ and $\varepsilon - \delta \leq \tau\sigma^2$. Then for the sample size N satisfying

$$N \geq \beta^{-1} \ln(2/\alpha) \text{ and } N \geq \frac{8\sigma^2}{(\varepsilon-\delta)^2}\left[n \ln\left(\frac{8\varrho LD}{\varepsilon-\delta}\right) + \ln\left(\frac{2}{\alpha}\right)\right], \tag{4.4}$$

it follows that

$$\mathsf{Prob}\big(\hat{\mathcal{S}}_N^\delta \subset \mathcal{S}^\varepsilon\big) \geq 1 - \alpha. \tag{4.5}$$

In particular, if in (4.2) the function $\kappa(\xi) \equiv L$, i.e., the Lipschitz constant of $F(\cdot,\xi)$ does not depend on ξ, then the first condition in the sample estimate (4.4) can be omitted and the constant σ^2 can be replaced by the estimate $2L^2D^2$. The assertion (4.5) of the above theorem means that if $\bar{x} \in \mathcal{X}$ is a δ-optimal solution of the SAA problem with the sample size N satisfying (4.4), then $\bar{x}$ is an ε-optimal solution of the true problem with probability $\geq 1-\alpha$. That is, by solving the SAA problem with accuracy $\delta < \varepsilon$, say $\delta := \varepsilon/2$, we are guaranteed with confidence $1-\alpha$ that we solve the true problem with accuracy ε. Similar estimates of the sample size can be obtained by using theory of Vapnik-Chervonenkis (VC) dimension (cf., [21]).

The above estimate of the sample size is theoretical and typically is too conservative for practical applications. Nevertheless it leads to several important conclusions. From this point of view the number of scenarios in a formulation of the true problem is irrelevant and can be infinite, while the computational difficulty is influenced by variability of the objective function which, in a sense, measured by the constant σ^2. It also suggests that the required sample size is proportional to ε^{-2}. Such dependence of the sample size on required accuracy is typical for Monte Carlo sampling methods and cannot be changed. Similar rates of convergence can be derived for the optimal value of the SAA problem. Central Limit Theorem type result of Theorem 2.1 confirms this from another point of view. In some situations quasi-Monte Carlo methods can enhance rates of convergence (cf., [6]), but in principle it is impossible to evaluate multidimensional integrals with a high accuracy. On the other hand dependence on the confidence $1-\alpha$ is logarithmic, e.g., increasing the confidence say from 90% to 99.99% does not require a big change of the sample size.

For well conditioned problems it is possible to derive better rates of convergence. It is said that a γ-order growth condition, with $\gamma \geq 1$, holds for the true problem if its set $\mathcal{S}$ of optimal solutions is nonempty and there is constant $c > 0$ such that

$$f(x) \geq \vartheta^* + c[\text{dist}(x,\mathcal{S})]^\gamma \tag{4.6}$$

for all $x \in \mathcal{X}$ in a neighborhood of $\mathcal{S}$. Of interest is the growth condition of order $\gamma = 1$ and $\gamma = 2$. If $\mathcal{S} = \{\bar{x}\}$ is a singleton and the first-order growth condition holds, the optimal solution $\bar{x}$ is referred to as *sharp*. For convex problems satisfying the second order growth condition the sample size estimate becomes of order $O(\varepsilon^{-1})$. In convex case of sharp optimal solution $\bar{x}$ the convergence is finite, in the sense that w.p.1 for N large enough the SAA problem has unique optimal solution $\bar{x}$ coinciding with the optimal solution of the true problem and, moreover, the probability of this event approaches one exponentially fast with increase of N (see [20, p.190] for a discussion and exact formulation).

5. Multistage Complexity

Consider the multistage setting of section 3. Recall that the optimal value ϑ^* of the multistage problem (1.3) is given by the optimal value of the problem (3.3) and $\mathcal{Q}_2(x_1) = \mathbb{E}[Q_2(x_1, \xi_2)]$. Similarly to the static case we say that $\bar{x}_1 \in \mathcal{X}_1$ is an ε-optimal solution of the first stage of the true problem (1.3) if $f_1(\bar{x}_1) + \mathcal{Q}_2(\bar{x}_1) \le \vartheta^* + \varepsilon$. Suppose for the moment that $T = 3$. Then under regularity conditions similar to the static case it is possible to derive the following estimate of the sample sizes N_1 and N_2 needed to solve the first stage problem with a given accuracy $\varepsilon > 0$ and confidence $1 - \alpha$ while solving the SAA problem with accuracy, say, $\varepsilon/2$ (see [20, section 5.8.2] for technical details).

> *For constants $\varepsilon > 0$ and $\alpha \in (0,1)$ and sample sizes N_1 and N_2 satisfying*
>
> $$\left[\tfrac{O(1)D_1L_1}{\varepsilon}\right]^{n_1} \exp\left\{-\tfrac{O(1)N_1\varepsilon^2}{\sigma_1^2}\right\} + \left[\tfrac{O(1)D_2L_2}{\varepsilon}\right]^{n_2} \exp\left\{-\tfrac{O(1)N_2\varepsilon^2}{\sigma_2^2}\right\} \le \alpha, \tag{5.1}$$
>
> *we have that any $(\varepsilon/2)$-optimal solution of the first stage of the SAA problem is an ε-optimal solution of the first stage of the true problem with probability at least $1 - \alpha$.* Here $O(1)$ denotes a generic constant independent of the data and σ_1, σ_2, D_1, D_2 and L_1, L_2 are certain analogues of the constants of the estimate (4.4).

In particular suppose that $N_1 = N_2$ and let $n := \max\{n_1, n_2\}$, $L := \max\{L_1, L_2\}$, $D := \max\{D_1, D_2\}$. Then (5.1) becomes

$$N_1 \ge \frac{O(1)\sigma^2}{\varepsilon^2}\left[n \ln\left(\frac{O(1)LD}{\varepsilon}\right) + \ln\left(\frac{1}{\alpha}\right)\right]. \tag{5.2}$$

The above estimate looks similar to the estimate (4.4) of the two stage program. Note, however, that in the present case of three stage program the total number of scenarios of the SAA problem is $N = N_1^2$. This analysis can be extended to a larger number of stages with the conclusion that the total number of scenarios needed to solve the true problem with a given accuracy grows *exponentially* with increase of the number T of stages. Another way of putting this is that the number of scenarios needed to solve T-stage problem (1.3) would grow as $O(\varepsilon^{-2(T-1)})$ with decrease of the error level $\varepsilon > 0$. This indicates that from the point of view of the number of scenarios, complexity of multistage programming problems grows exponentially with the number of stages. Furthermore, as it was pointed in the Introduction, even if the SAA problem can be solved, its solution does not define a policy for the true problem and of use may be only the computed first stage solution. There are even deeper reasons to believe that (even linear) multistage stochastic programs are computationally intractable (cf., [19]). This does not mean, of course, that some specific classes of multistage stochastic programs cannot be solved efficiently.

6. Approximations of Multistage Stochastic Programs

If multistage stochastic programming problems cannot be solve to optimality, one may think about approximations. There are several possible approaches to trying to solve multistage stochastic programs approximately. One approach is to reduce dynamic setting to a static case. Suppose that we can identify a parametric family of policies $\bar{x}_t(\xi_{[t]}, \theta_t)$, $t = 1, ..., T$, depending on a finite number of parameters $\theta_t \in \Theta_t \subset \mathbb{R}^{q_t}$, and such that these policies are feasible for all parameter values. That is, for all $\theta_t \in \Theta_t$, $t = 1, ..., T$, it holds that $\bar{x}_1(\theta_1) \in \mathcal{X}_1$ and $\bar{x}_t(\xi_{[t]}, \theta_t) \in \mathcal{X}_t\big(\bar{x}_{t-1}(\xi_{[t-1]}, \theta_{t-1}), \xi_t\big)$, $t = 2, ..., T$, w.p.1. Consider the following stochastic program

$$\begin{array}{cl} \underset{\theta_1,\dots,\theta_T}{\text{Min}} & f_1\big(\bar{x}_1(\theta_1)\big) + \mathbb{E}\left[\sum_{t=2}^T f_t\big(\bar{x}_t(\xi_{[t]}, \theta_t), \xi_t\big)\right] \\ \text{s.t.} & \theta_t \in \Theta_t,\ t = 1, ..., T. \end{array} \tag{6.1}$$

The above problem (6.1) is a (static) stochastic problem of the form (2.1) and could be solved, say by the SAA method, provided that the sets Θ_t are defined in a computationally accessible way. Of course, quality of an obtained solution $\bar{x}_t(\xi_{[t]}, \theta_t^*)$, $t = 1, ..., T$, viewed as a solution of the original multistage problem (1.3), depends on a successful choice of the parametric family.

Suppose, for example, that we have a finite family of feasible policies $\big\{x_t^k(\xi_{[t]}),\ t = 1, ..., T\big\}$, $k = 1, ..., K$. Suppose, further, that the multifunctions $\mathcal{X}_t(\cdot, \xi_t)$ are convex, i.e., the set $\mathcal{X}_1$ is convex and for a.e. ξ_t and all x_{t-1}, x'_{t-1} and $\tau \in [0, 1]$ it holds that

$$\tau\mathcal{X}_t(x_{t-1}, \xi_t) + (1-\tau)\mathcal{X}_t(x'_{t-1}, \xi_t) \subset \mathcal{X}_t\big(\tau x_{t-1} + (1-\tau)x'_{t-1}, \xi_t\big),\ t = 2, ..., T.$$

Then any convex combination

$$\bar{x}_t(\xi_{[t]}, \theta) := \sum_{k=1}^K \theta_k x_t^k(\xi_{[t]}),\ t = 1, ..., T,$$

where $\theta = (\theta_1, ..., \theta_K) \in \Delta_K$ with Δ_K being K-dimensional simplex, of these policies is feasible. This approach with several examples was discussed in [8].

As another example consider linear multistage stochastic programs with fixed recourse. That is, assume the setting of (1.4)–(1.5) with only the right hand sides vectors b_t, $t = 2, ..., T$, being random. Moreover, for the sake of simplicity assume that the data process $b_1, ..., b_T$, is stagewise independent with distribution of random vector b_t supported on set Ξ_t, $t = 2, ..., T$. Motivated by its success in robust optimization it was suggested in [19] to use affine decision rules. That is, consider policies of the form

$$\bar{x}_t = \phi_t + \sum_{\tau=2}^{t} \Phi_{t\tau} b_\tau, \quad t = 2, ..., T, \tag{6.2}$$

depending on parameters – vectors ϕ_t and matrices $\Phi_{t\tau}$. The feasibility constraints here take the form

$$\begin{array}{l} A_1 x_1 \le b_1, \ B_2 x_1 + A_2\big(\phi_2 + \Phi_{22} b_2\big) \le b_2, \\ B_t\big(\phi_{t-1} + \sum_{\tau=2}^{t-1} \Phi_{t-1,\tau} b_\tau\big) + A_t\big(\phi_t + \sum_{\tau=2}^{t} \Phi_{t\tau} b_\tau\big) \le b_t \ \ t = 3, ..., T, \end{array} \tag{6.3}$$

and should hold for every $b_t \in \Xi_t$, $t = 2, ..., T$ (we can pass here from the condition "for a.e." to "for every" by continuity arguments). The system (6.3), defining feasible parameters of the policy (6.2), involves an infinite number of linear constraints. In case the sets Ξ_t are polyhedral, defined by a finite number of linear constraints, it is possible to handle the semi-infinite system (6.3) in a computationally efficient way (cf., [19]).

An alternative approach to solving multistage program (1.3) is to approximate dynamic programming equations (3.1). One such approach can be described as follows. Consider the linear setting (1.4)–(1.5) and assume that the stagewise independence condition holds. In that case the cost-to-go functions $\mathcal{Q}_t(x_{t-1})$, $t = 2, ..., T$, are convex and do not depend on the random data. Consider the corresponding SAA problem based on (independent) samples $\xi_t^1, ..., \xi_t^{N_{t-1}}$, $t = 2, ..., T$. By the above analysis we have (under mild regularity conditions) that if all sample sizes are of the same order, say all $N_t = M$, $t = 1, ..., T-1$, then in order to solve the first stage problem with accuracy $\varepsilon > 0$ we need M to be of order $O(\varepsilon^{-2})$. Of course, even for moderate values of M, say $M = 100$, the total number of scenarios $N = M^{T-1}$ quickly becomes astronomically large with increase of the number of stages. Therefore, instead of solving the corresponding linear programming problem involving all scenarios, one can try to approximate the cost-to-go functions of the SAA problem.

For a given set of samples of size $\mathcal{N} = (N_1, ..., N_{T-1})$, let $\tilde{\mathcal{Q}}_{t,\mathcal{N}}(x_{t-1})$, $t = 2, ..., T$, be cost-to-go functions of the SAA problem. These functions are convex piecewise linear and do not depend on realizations of scenarios from the SAA scenario tree. Suppose that we have a procedure for generating cutting (supporting) planes for the SAA cost-to-go functions. By taking maximum of respective collections of these cutting planes we can construct piecewise linear convex functions $\mathfrak{Q}_t(x_{t-1})$ approximating the SAA cost-to-go functions from below, i.e., $\tilde{\mathcal{Q}}_{t,\mathcal{N}}(\cdot) \ge \mathfrak{Q}_t(\cdot)$, $t = 2, ..., T$. These functions $\mathfrak{Q}_t(x_{t-1})$ and a feasible first stage solution $\bar{x}_1$ define the following policy:

$$\bar{x}_t \in \arg\min \left\{ \langle c_t, x_t \rangle + \mathfrak{Q}_{t+1}(x_t) : A_t x_t \le b_t - B_t \bar{x}_{t-1} \right\}, \ t = 2, ..., T, \tag{6.4}$$

with $\mathfrak{Q}_{T+1}(x_T) \equiv 0$ by definition. This policy can be applied to the true multistage problem and to its sample average approximation. In both cases the policy is feasible by the construction and hence its expected value gives an upper bound for the optimal value of the corresponding multistage program. The expected value of this policy can be estimated by sampling.

Since functions $\mathfrak{Q}_t(\cdot)$ are given as maximum of a finite number of affine functions, the optimization problems in the right hand side of (6.4) can be formulated as linear programming problems of reasonable sizes. It was suggested in [14] to generate trial decision points $\bar{x}_t$ using randomly generated sample paths in a forward step procedure of the form (6.4) and consequently to add cutting planes, computed at these trial decision points, to approximations $\mathfrak{Q}_t(\cdot)$ in a backward step procedure. The required cutting planes are constructed by solving duals of the linear programming problems associated with right hand side of (6.4). This algorithm, called Stochastic Dual Dynamic Programming (SDDP), became popular in energy planning procedures. It is possible to show that under mild regularity conditions, functions $\mathfrak{Q}_t(\cdot)$ converge w.p.1 to their counterparts $\tilde{\mathcal{Q}}_{t,\mathcal{N}}(\cdot)$ of the SAA problem, and hence policy (6.4) converges to an optimal policy of the SAA problem (cf., [15]). The convergence can be slow however.

> For two stage programs the SDDP algorithm becomes Kelley's cutting plane algorithm, [7]. Worst case analysis of Kelley's algorithm is discussed in [13, pp. 158-160], with an example of a problem where an ε-optimal solution cannot be obtained by this algorithm in less than $\left(\frac{1}{2\ln 2}\right) 1.15^{n-1} \ln(\varepsilon^{-1})$ calls of the oracle, i.e., the number of oracle calls grows exponentially with increase of the dimension n of the problem. It was also observed empirically that Kelley's algorithm could behave quite poorly in practice.

On the other hand, complexity of one run of the forward and backward steps of the SDDP algorithm grows linearly with increase of the number of stages and the algorithm produces a feasible and implementable policy.

7. Concluding Remarks

So far we discussed computational complexity from the point of view of the required number of scenarios. It should be remembered that a constructed SAA problem still needs to be solved by an appropriate deterministic algorithm. Consider for example the SAA problem associated with two stage linear problem (1.1). In order to compute a subgradient of the respective sample average function $\hat{Q}_N(x) = \frac{1}{N}\sum_{j=1}^{N} Q(x, \xi^j)$ at an iteration point of a subgradient type algorithmic procedure, one would need to solve N second stage problems together with their duals.

For *convex* (static) stochastic problems an alternative to the SAA approach is the Stochastic Approximation (SA) method going back to Robbins and Monro

[16]. The classical SA algorithm generates iterates for solving problem (2.1) by the formula

$$x_{j+1} = \Pi_{\mathcal{X}}\big(x_j - \gamma_j G(x_j, \xi^j)\big), \ \ j = 1, ..., \tag{7.1}$$

where $G(x, \xi) \in \partial_x F(x, \xi)$ is a subgradient of $F(x, \xi)$, $\Pi_{\mathcal{X}}$ is the metric projection onto the set $\mathcal{X}$ and $\gamma_j > 0$ are chosen stepsizes. The standard choice of the stepsizes is $\gamma_j = \theta/j$ for some constant $\theta > 0$. For an *optimal* choice of the constant θ the estimates of rates of convergence of this method are similar to the respective estimates of the SAA method. However, the method is very sensitive to the choice of the constant θ and often does not work well in practice. It is somewhat surprising that a robust version of the SA algorithm, taking its origins in the mirror descent method of Nemirovski and Yudin [11], can significantly outperform SAA based algorithms for certain classes of convex stochastic problems (cf., [12]).

Theoretical estimates of the form (4.4), of the required sample size, are too conservative for practical applications. In that respect we may refer to [10] and [9] for a discussion of computational methods for evaluating quality of solutions of the first stage of two stage stochastic problems.

References

[1] Araujo, A. and Giné, E., *The Central Limit Theorem for Real and Banach Valued Random Variables.* Wiley, New York, 1980.

[2] Beale, E.M.L., On minimizing a convex function subject to linear inequalities, *Journal of the Royal Statistical Society, Series B*, **17** (1955), 173–184.

[3] Ben-Tal, A., El Ghaoui, L. and Nemirovski, A., *Robust Optimization*, Princeton University Press, Princeton, 2009.

[4] Dantzig, G.B., Linear programming under uncertainty, *Management Science*, **1** (1955), 197–206.

[5] Dyer, M. and Stougie, L., Computational complexity of stochastic programming problems, *Mathematical Programming*, **106** (2006), 423–432.

[6] Homem-de Mello, T., On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling, *SIAM J. Optim.*, **19** (2008), 524–551.

[7] Kelley, J.E., The cutting-plane method for solving convex programs, *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 703–712.

[8] Koivu, M. and Pennanen, T, Galerkin methods in dynamic stochastic programming, *Optimization*, to appear.

[9] Lan, G., Nemirovski, A. and Shapiro, A., Validation analysis of mirror descent stochastic approximation method, *E-print available at:* `http://www.optimization-online.org`, 2008.

[10] Mak, W.K., Morton, D.P. and Wood, R.K., Monte Carlo bounding techniques for determining solution quality in stochastic programs, *Operations Research Letters*, **24** (1999), 47–56.

[11] Nemirovski, A. and Yudin, D., *Problem Complexity and Method Efficiency in Optimization*, Wiley-Intersci. Ser. Discrete Math. 15, John Wiley, New York, 1983.

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

[13] Nesterov, Yu., *Introductory Lectures on Convex Optimization*, Kluwer, Boston, 2004.

[14] Pereira, M.V.F. and Pinto, L.M.V.G., Multi-stage stochastic optimization applied to energy planning, *Mathematical Programming*, **52** (1991), 359–375.

[15] Philpott, A.B. and Guan, Z., On the convergence of stochastic dual dynamic programming and related methods, *Operations Research Letters*, **36** (2008), 450–455.

[16] Robbins, H. and Monro, S., A stochastic approximation method. *Annals of Math. Stat.*, **22** (1951), 400–407.

[17] Shapiro, A., Asymptotic analysis of stochastic programs. *Annals of Operations Research*, **30** (1991), 169–186.

[18] Shapiro, A., Monte Carlo approach to stochastic programming. In B.A. Peters, J.S. Smith, D.J. Medeiros and M.W. Rohrer, editors, *Proceedings of the 2001 Winter Simulation Conference*, pp. 428–431, 2001.

[19] Shapiro, A. and Nemirovski, A., On complexity of stochastic programming problems, in: *Continuous Optimization: Current Trends and Applications*, pp. 111–144, V. Jeyakumar and A.M. Rubinov (Eds.), Springer, 2005.

[20] Shapiro, A., Dentcheva, D. and Ruszczyński, A., *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia, 2009.

[21] Vidyasagar, M., Randomized algorithms for robust controller synthesis using statistical learning theory, *Automatica*, **37** (2001), 1515–1528.