Stochastic Approximation Approach to Stochastic Programming

Anatoli Juditsky[*] Guanghui Lan[†] Arkadi Nemirovski[‡] Alexander Shapiro [§]

Abstract. In this paper we consider optimization problems where the objective function is given in a form of the expectation. A basic difficulty of solving such stochastic optimization problems is that the involved multidimensional integrals (expectations) cannot be computed with high accuracy. The aim of this paper is to compare two computational approaches based on Monte Carlo sampling techniques, namely, the Stochastic Approximation (SA) and the Sample Average Approximation (SAA) methods. Both approaches, the SA and SAA methods, have a long history. Current opinion is that the SAA method can efficiently use a specific (say linear) structure of the considered problem, while the SA approach is a crude subgradient method which often performs poorly in practice. We intend to demonstrate that a properly modified SA approach can be competitive and even significantly outperform the SAA method for a certain class of convex stochastic problems. We extend the analysis to the case of convex-concave stochastic saddle point problems, and present (in our opinion highly encouraging) results of numerical experiments.

Key words: stochastic approximation, sample average approximation method, stochastic programming, Monte Carlo sampling, complexity, saddle point, minimax problems, mirror descent algorithm

[*] LJK, Université J. Fourier, B.P. 53, 38041 Grenoble Cedex 9, France, Anatoli.Juditsky@imag.fr

[†] Georgia Institute of Technology, Atlanta, Georgia 30332, USA, glan@isye.gatech.edu, research of this author was partially supported by NSF award CCF-0430644 and ONR award N00014-05-1-0183.

[‡] Georgia Institute of Technology, Atlanta, Georgia 30332, USA, nemirovs@isye.gatech.edu, research of this author was partly supported by the NSF award DMI-0619977.

[§] Georgia Institute of Technology, Atlanta, Georgia 30332, USA, ashapiro@isye.gatech.edu, research of this author was partly supported by the NSF awards DMS-0510324 and DMI-0619977.

1 Introduction

In this paper we consider the following optimization problem

$$\min_{x\in X}\big\{f(x):=\mathbb{E}[F(x,\xi)]\big\}. \tag{1.1}$$

Here $X\subset\mathbb{R}^n$ is a nonempty bounded closed convex set, ξ is a random vector whose probability distribution P is supported on set $\Xi\subset\mathbb{R}^d$ and $F: X\times\Xi\to\mathbb{R}$. We assume that for every $\xi\in\Xi$ the function $F(\cdot,\xi)$ is convex on X, and that the expectation

$$\mathbb{E}[F(x,\xi)]=\textstyle\int_\Xi F(x,\xi)dP(\xi) \tag{1.2}$$

is well defined and finite valued for every $x\in X$. It follows that function $f(\cdot)$ is convex and finite valued on X. Moreover, we assume that $f(\cdot)$ is continuous on X. Of course, continuity of $f(\cdot)$ follows from convexity if $f(\cdot)$ is finite valued and convex on a neighborhood of X. With these assumptions, (1.1) becomes a convex programming problem.

A basic difficulty of solving stochastic optimization problem (1.1) is that the multidimensional integral (expectation) (1.2) cannot be computed with a high accuracy for dimension d, say, greater than 5. The aim of this paper is to compare two computational approaches based on Monte Carlo sampling techniques, namely, the *Stochastic Approximation* (SA) and the *Sample Average Approximation* (SAA) methods. To this end we make the following assumptions.

(A1) It is possible to generate an iid sample $\xi_1,\xi_2,...,$ of realizations of random vector ξ.

(A2) There is a mechanism (an oracle) which for every given $x\in X$ and $\xi\in\Xi$ returns value $F(x,\xi)$ and a *stochastic subgradient* – a vector $\mathsf{G}(x,\xi)$ such that $\mathsf{g}(x):=\mathbb{E}[\mathsf{G}(x,\xi)]$ is well defined and is a subgradient of $f(\cdot)$ at x, i.e., $\mathsf{g}(x)\in\partial f(x)$.

Recall that if $F(\cdot,\xi)$, $\xi\in\Xi$, is convex and $f(\cdot)$ is finite valued in a neighborhood of a point x, then (cf., Strassen [18])

$$\partial f(x)=\mathbb{E}\left[\partial_x F(x,\xi)\right]. \tag{1.3}$$

In that case we can employ a measurable selection $\mathsf{G}(x,\xi)\in\partial_x F(x,\xi)$ as a stochastic subgradient. At this stage, however, this is not important, we shall see later other relevant ways for constructing stochastic subgradients.

Both approaches, the SA and SAA methods, have a long history. The SA method is going back to the pioneering paper by Robbins and Monro [13]. Since then stochastic approximation algorithms became widely used in stochastic optimization and, due to especially low demand for computer memory, in signal processing (cf., [3] and references therein). In the classical analysis of the SA algorithm (it apparently goes back to the works [4] and [14]) it is assumed that f is twice continuously differentiable and strongly convex, and in the case when the minimizer of f belongs to the interior of X, exhibits asymptotically optimal rate of convergence $\mathbb{E}[f(x_t)-f_*]=O(1/t)$ (here x_t is t-th iterate and f_* is the minimal value of $f(x)$ over $x\in X$). This algorithm, however, is very sensitive to a choice of the respective stepsizes. The difficult to implement "asymptotically optimal" stepsize policy can be very bad in the beginning, so that the algorithm often performs poorly in practice.

An important improvement of the SA method was developed by B. Polyak [11, 12], where longer stepsizes were suggested with consequent averaging of the obtained iterates. Under the outlined "classical" assumptions, the resulting algorithm exhibits the same optimal $O(1/t)$ asymptotical convergence rate, while using an easy to implement and "robust" stepsize policy. It should be mentioned that the main ingredients of Polyak's scheme – long steps and averaging – were, in a different form, proposed already in [9] for the case of problems (1.1) with general type Lipschitz continuous convex objectives and for convex-concave saddle point problems. The algorithms from [9] exhibit, in a non-asymptotical fashion, the unimprovable in the general convex case $O(1/\sqrt{t})$-rate of convergence. For a summary of early results in this direction, see [10].

The SAA approach was used by many authors in various contexts under different names. Its basic idea is rather simple: generate a (random) sample $\xi_1, ..., \xi_N$, of size N, and approximate the "true" problem (1.1) by the sample average problem

$$\min_{x \in X} \left\{ \hat{f}_N(x) := N^{-1} \sum_{j=1}^{N} F(x, \xi_j) \right\}. \tag{1.4}$$

Note that the SAA method is not an algorithm, the obtained SAA problem (1.4) still has to be solved by an appropriate numerical procedure. Recent theoretical studies (cf., [6, 16, 17]) and numerical experiments (see, e.g., [7, 8, 19]) show that the SAA method coupled with a good (deterministic) algorithm could be reasonably efficient for solving certain classes of two stage stochastic programming problems. On the other hand classical SA type numerical procedures typically performed poorly for such problems. We intend to demonstrate in this paper that a properly modified SA approach can be competitive and even significantly outperform the SAA method for a certain class of stochastic problems.

The rest of this paper is organized as follows. In Section 2 we focus on the theory of SA as applied to problem (1.1). We start with outlining the (relevant to our goals part of the) classical "$O(1/t)$ SA theory" (Section 2.1) along with its "$O(1/\sqrt{t})$" modifications (Section 2.2). Well known and simple results presented in these sections pave road to our main developments carried out in Section 2.3. In Section 3 we extend the constructions and results of Section 2.3 to the case of convex-concave stochastic saddle point problem. In concluding Section 4 we present results (in our opinion, highly encouraging) of numerical experiments with the SA algorithm (Sections 2.3 and 3) applied to large-scale stochastic convex minimization and saddle point problems. Finally, some technical proofs are given in the Appendix.

Throughout the paper, we use the following notation. By $\|x\|_p$ we denote the ℓ_p norm of vector $x \in \mathbb{R}^n$, in particular, $\|x\|_2 = \sqrt{x^T x}$ denotes the Euclidean norm. By Π_X we denote the metric projection operator onto the set X, that is $\Pi_X(x) = \arg\min_{x' \in X} \|x - x'\|_2$. Note that Π_X is a contraction operator, i.e.,

$$\|\Pi_X(x') - \Pi_X(x)\|_2 \leq \|x' - x\|_2, \ \ \forall x', x \in \mathbb{R}^n. \tag{1.5}$$

By $O(1)$ we denote a generic constant independent of the data. The notation $\lfloor a \rfloor$ stands for the largest integer less than or equal to $a \in \mathbb{R}$. Unless stated otherwise all relations between random variables are supposed to hold almost surely.

2 Stochastic Approximation, Basic Theory

In this section we discuss theory and implementations of the stochastic approximation (SA) approach to the minimization problem (1.1).

2.1 Classical SA Algorithm

The classical SA algorithm solves problem (1.1) by mimicking the simplest subgradient descent method. That is, for chosen $x_1 \in X$ and a sequence $\gamma_j > 0$, $j = 1, ...,$ of stepsizes, it generates the iterates by the formula

$$x_{j+1} := \Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big). \tag{2.1}$$

Of course, the crucial question of that approach is how to choose the stepsizes γ_j. Let $\bar{x}$ be an optimal solution of problem (1.1). Note that since the set X is compact and $f(x)$ is continuous, problem (1.1) has an optimal solution. Note also that the iterate $x_j = x_j(\xi_{[j-1]})$ is a function of the history $\xi_{[j-1]} := (\xi_1, ..., \xi_{j-1})$ of the generated random process and hence is random.

Denote $A_j := \frac{1}{2}\|x_j - \bar{x}\|_2^2$ and $a_j := \mathbb{E}[A_j] = \frac{1}{2}\mathbb{E}\big[\|x_j - \bar{x}\|_2^2\big]$. By using (1.5) and since $\bar{x} \in X$ and hence $\Pi_X(\bar{x}) = \bar{x}$, we can write

$$\begin{array}{rl} A_{j+1} & = \frac{1}{2}\big\|\Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big) - \bar{x}\big\|_2^2 \\ & = \frac{1}{2}\big\|\Pi_X\big(x_j - \gamma_j \mathsf{G}(x_j, \xi_j)\big) - \Pi_X(\bar{x})\big\|_2^2 \\ & \leq \frac{1}{2}\big\|x_j - \gamma_j \mathsf{G}(x_j, \xi_j) - \bar{x}\big\|_2^2 \\ & = A_j + \frac{1}{2}\gamma_j^2\|\mathsf{G}(x_j, \xi_j)\|_2^2 - \gamma_j (x_j - \bar{x})^T \mathsf{G}(x_j, \xi_j). \end{array} \tag{2.2}$$

We also have

$$\begin{array}{rcl} \mathbb{E}\big[(x_j - \bar{x})^T \mathsf{G}(x_j, \xi_j)\big] & = & \mathbb{E}_{\xi_{[j-1]}}\big\{\mathbb{E}_{\xi_j}\big[(x_j - \bar{x})^T \mathsf{G}(x_j, \xi_j)\big]\big\} \\ & = & \mathbb{E}_{\xi_{[j-1]}}\big\{(x_j - \bar{x})^T \mathbb{E}\big[\mathsf{G}(x_j, \xi_j)\big]\big\} \\ & = & \mathbb{E}\left[(x_j - \bar{x})^T \mathsf{g}(x_j)\right]. \end{array} \tag{2.3}$$

Therefore, by taking expectation of both sides of (2.2) we obtain

$$a_{j+1} \leq a_j - \gamma_j \mathbb{E}\left[(x_j - \bar{x})^T \mathsf{g}(x_j)\right] + \tfrac{1}{2}\gamma_j^2 M^2, \tag{2.4}$$

where

$$M^2 := \sup_{x \in X} \mathbb{E}\left[\|\mathsf{G}(x, \xi)\|_2^2\right]. \tag{2.5}$$

We assume that the above constant M is finite.

Suppose, further, that the expectation function $f(x)$ is differentiable and strongly convex on X, i.e., there is constant $c > 0$ such that

$$f(x') \geq f(x) + (x' - x)^T \nabla f(x) + \tfrac{1}{2}c\|x' - x\|_2^2, \;\; \forall x', x \in X,$$

or equivalently that

$$(x' - x)^T(\nabla f(x') - \nabla f(x)) \geq c\|x' - x\|_2^2, \;\; \forall x', x \in X. \tag{2.6}$$

Note that strong convexity of $f(x)$ implies that the minimizer $\bar{x}$ is unique. By optimality of $\bar{x}$ we have that

$$(x - \bar{x})^T \nabla f(\bar{x}) \geq 0, \;\; \forall x \in X,$$

which together with (2.6) implies that

$$\mathbb{E}\left[(x_j - \bar{x})^T \nabla f(x_j)\right] \geq \mathbb{E}\left[(x_j - \bar{x})^T(\nabla f(x_j) - \nabla f(\bar{x}))\right] \geq c\mathbb{E}\left[\|x_j - \bar{x}\|_2^2\right] = 2ca_j.$$

Therefore it follows from (2.4) that

$$a_{j+1} \le (1 - 2c\gamma_j)a_j + \tfrac{1}{2}\gamma_j^2 M^2. \tag{2.7}$$

Let us take stepsizes $\gamma_j = \theta/j$ for some constant $\theta > 1/(2c)$. Then by (2.7) we have

$$a_{j+1} \le (1 - 2c\theta/j)a_j + \tfrac{1}{2}\theta^2 M^2/j^2,$$

and by induction

$$a_j \le \kappa/j, \tag{2.8}$$

where

$$\kappa := \max\left\{\tfrac{1}{2}\theta^2 M^2 (2c\theta - 1)^{-1}, a_1\right\}.$$

Suppose, further, that $\bar{x}$ is an *interior* point of X and $\nabla f(x)$ is Lipschitz continuous, i.e., there is constant $L > 0$ such that

$$\|\nabla f(x') - \nabla f(x)\|_2 \le L\|x' - x\|_2, \ \ \forall x', x \in X. \tag{2.9}$$

Then

$$f(x) \le f(\bar{x}) + \tfrac{1}{2}L\|x - \bar{x}\|_2^2, \ \ \forall x \in X, \tag{2.10}$$

and hence

$$\mathbb{E}\big[f(x_j) - f(\bar{x})\big] \le La_j \le L\kappa/j. \tag{2.11}$$

Under the specified assumptions, it follows from (2.10) and (2.11), respectively, that after t iterations the expected error of the current solution is of order $O(t^{-1/2})$ and the expected error of the corresponding objective value is of order $O(t^{-1})$, provided that $\theta > 1/(2c)$. We have arrived at the $O(t^{-1})$-rate of convergence mentioned in the Introduction. Note, however, that the result is highly sensitive to our a priori information on c. What would happen if the parameter c of strong convexity is overestimated? As a simple example consider $f(x) = x^2/10$, $X = [-1, 1] \subset \mathbb{R}$ and assume that there is no noise, i.e., $F(x, \xi) \equiv f(x)$. Suppose, further, that we take $\theta = 1$ (i.e., $\gamma_j = 1/j$), which will be the optimal choice for $c = 1$, while actually here $c = 0.2$. Then the iteration process becomes

$$x_{j+1} = x_j - f'(x_j)/j = \left(1 - \tfrac{1}{5j}\right)x_j,$$

and hence starting with $x_1 = 1$,

$$\begin{aligned} x_j &= \textstyle\prod_{s=1}^{j-1}\left(1 - \frac{1}{5s}\right) = \exp\left\{-\sum_{s=1}^{j-1}\ln\left(1 + \frac{1}{5s-1}\right)\right\} > \exp\left\{-\sum_{s=1}^{j-1}\frac{1}{5s-1}\right\} \\ &> \exp\left\{-\left(0.25 + \int_1^{j-1}\frac{1}{5t-1}dt\right)\right\} > \exp\left\{-0.25 + 0.2\ln 1.25 - \tfrac{1}{5}\ln j\right\} > 0.8j^{-1/5}. \end{aligned}$$

That is, the convergence is extremely slow. For example for $j = 10^9$ the error of the iterated solution is greater than 0.015. On the other hand for the optimal stepsize factor of $\gamma = 1/c = 5$, the optimal solution $\bar{x} = 0$ is found in one iteration.

2.2 Robust SA Approach

The results of this section go back to [9] and [10].

Let us look again at the basic estimate (2.4). By convexity of $f(x)$ we have that for any x, $f(x) \geq f(x_j) + (x - x_j)^T \mathsf{g}(x_j)$, and hence

$$\mathbb{E}\big[(x_j - \bar{x})^T \mathsf{g}(x_j)\big] \geq \mathbb{E}\big[f(x_j) - f(\bar{x})\big] = \mathbb{E}\big[f(x_j)\big] - f(\bar{x}).$$

Together with (2.4) this implies

$$\gamma_j \mathbb{E}\big[f(x_j) - f(\bar{x})\big] \leq a_j - a_{j+1} + \tfrac{1}{2}\gamma_j^2 M^2.$$

It follows that

$$\sum_{t=1}^{j} \gamma_t \mathbb{E}\big[f(x_t) - f(\bar{x})\big] \leq \sum_{t=1}^{j} [a_t - a_{t+1}] + \tfrac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2 \leq a_1 + \tfrac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2, \tag{2.12}$$

and hence

$$\mathbb{E}\left[\sum_{t=1}^{j} \nu_t f(x_t) - f(\bar{x})\right] \leq \frac{a_1 + \frac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2}{\sum_{t=1}^{j} \gamma_t}, \tag{2.13}$$

where $\nu_t := \frac{\gamma_t}{\sum_{i=1}^{j} \gamma_i}$ (note that $\sum_{t=1}^{j} \nu_t = 1$). Consider points

$$\tilde{x}_j := \sum_{t=1}^{j} \nu_t x_t. \tag{2.14}$$

By convexity of $f(x)$ we have $f(\tilde{x}_j) \leq \sum_{t=1}^{j} \nu_t f(x_t)$, and since $\tilde{x}_j \in X$, by optimality of $\bar{x}$ we have that $f(\tilde{x}_j) \geq f(\bar{x})$. Thus, by (2.13),

$$0 \leq \mathbb{E}\left[f(\tilde{x}_j) - f(\bar{x})\right] \leq \frac{a_1 + \frac{1}{2}M^2 \sum_{t=1}^{j} \gamma_t^2}{\sum_{t=1}^{j} \gamma_t}. \tag{2.15}$$

Let us suppose for the moment that the number of iterations of the method is fixed in advance, say equal to N. In this case one can use a *constant stepsize strategy*, i.e., choose $\gamma_t \equiv \gamma$ for $t = 1, ..., N$. For this choice of γ_t we obtain immediately from (2.15) that the obtained approximate solution

$$\tilde{x}_N = N^{-1} \sum_{t=1}^{N} x_t, \tag{2.16}$$

satisfies:

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \leq \frac{a_1}{\gamma N} + \frac{M^2 \gamma}{2}. \tag{2.17}$$

Let us denote $D_X := \max_{x \in X} \|x - x_1\|_2$. Then $a_1 \leq D_X^2/2$, and taking

$$\gamma := \frac{D_X}{M\sqrt{N}}, \tag{2.18}$$

we achieve

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \leq \frac{D_X M}{\sqrt{N}}. \tag{2.19}$$

Discussion. We conclude that the expected error of *Robust SA* algorithm (2.1),(2.16), with constant stepsize strategy (2.18), after N iterations is $O(N^{-1/2})$ in our setting. Of course, this is worse than the rate $O(N^{-1})$ for the classical SA algorithm when the objective function $f(x)$ is strongly convex. However, the error bound (2.19) is guaranteed wether the function $f(x)$ is strongly convex on X or not. Note also that it follows from (2.17) that the rate $O(N^{-1/2})$ is guaranteed for the constant stepsize of the form $\gamma := \theta/\sqrt{N}$ for any choice of the constant $\theta > 0$. This explains the adjective *Robust* in the name of the algorithm.

In applications it can be convenient to construct an approximate solution $\tilde{x}_N$ which is the average of only part of the trajectory $x_1, ..., x_N$. For instance, let for some integer $\ell \in \{1, ..., N\}$,

$$\tilde{x}_j := \frac{1}{\lfloor N/\ell \rfloor} \sum_{t=N-\lfloor N/\ell \rfloor+1}^{N} x_t. \tag{2.20}$$

If we sum in (2.12) between $N - \lfloor N/\ell \rfloor$ and N (instead of summing from 1 to N) we easily get

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \leq \frac{\overline{D}_X M(\ell+1)}{2\sqrt{N}}, \tag{2.21}$$

where $\overline{D}_X := \max_{x',x\in X} \|x' - x\|_2$.

Of course, the constant stepsize strategy is not the only possible one. For instance, let $\gamma_j = \theta j^{-1/2}$, $j = 1, 2, ...$, with $\theta > 0$ and let $\tilde{x}_j$ be defined as follows:

$$\tilde{x}_j = \left(\sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t \right)^{-1} \sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t x_t,$$

for some integer $2 \leq \ell \leq j$ and $j \geq 2$. Then

$$\mathbb{E}\left[f(\tilde{x}_j) - f(\bar{x})\right] \leq O(1) \frac{\ell[\overline{D}_X^2 + M^2\theta^2]}{\theta\sqrt{j}}. \tag{2.22}$$

Note that $(\overline{D}_X^2 + M^2\theta^2)/\theta$ attains its minimum at $\theta = \overline{D}_X/M$. For that choice of θ the estimate (2.22) becomes

$$\mathbb{E}\left[f(\tilde{x}_j) - f(\bar{x})\right] \leq O(1) \frac{\ell \overline{D}_X M}{\sqrt{j}}. \tag{2.23}$$

2.3 Mirror Descent SA Method

In this section we develop a substantial generalization of the robust SA approach (a very rudimentary form of this generalization can be found in [10], from where, in particular, the name “Mirror Descent” originates). Let $\|\cdot\|$ be a (general) norm on $\mathbb{R}^n$ and $\|x\|_* = \sup_{\|y\|\leq 1} y^T x$ be its dual norm. We say that a function $\omega : X \to \mathbb{R}$ is a *distance generating function* modulus $\alpha > 0$ with respect to $\|\cdot\|$, if ω is convex and continuous on X, the set

$$X^o := \left\{ x \in X : \text{there exists } p \in \mathbb{R}^n \text{ such that } x \in \arg\min_{u\in X}[p^T u + \omega(u)] \right\}$$

is convex (note that X^o always contains the relative interior of X), and restricted to X^o, ω is continuously differentiable and strongly convex with parameter α with respect to $\|\cdot\|$, i.e.,

$$(x'-x)^T(\nabla\omega(x')-\nabla\omega(x)) \geq \alpha\|x'-x\|^2, \ \ \forall x', x \in X^o. \tag{2.24}$$

An example of distance generating function (modulus 1 with respect to $\|\cdot\|_2$) is $\omega(x) := \frac{1}{2}\|x\|_2^2$. For that choice of $\omega(\cdot)$ we have that $X^o = X$,

$$\Pi_X(x-y) = \arg\min_{z\in X}\left\{\frac{\|z-x\|_2^2}{2} + y^T(z-x)\right\}$$

and

$$\frac{\|z-x\|_2^2}{2} = \omega(z) - [\omega(x) + \nabla\omega(x)^T(z-x)].$$

Let us define function $V : X^o \times X \to \mathbb{R}_+$ as follows

$$V(x,z) := \omega(z) - [\omega(x) + \nabla\omega(x)^T(z-x)]. \tag{2.25}$$

In what follows we shall refer to $V(\cdot,\cdot)$ as *prox-function* associated with distance generating function $\omega(x)$. Note that $V(x,\cdot)$ is nonnegative and is strongly convex modulus α with respect to the norm $\|\cdot\|$. Let us define *prox mapping* $P_x : \mathbb{R}^n \to X^o$, associated with ω and a point $x \in X^o$, viewed as a parameter, as follows:

$$P_x(y) := \arg\min_{z\in X}\left\{y^T(z-x) + V(x,z)\right\}. \tag{2.26}$$

For $\omega(x) = \frac{1}{2}\|x\|_2^2$ we have that $P_x(y) = \Pi_X(x-y)$. Let us observe that the minimum in the right hand side of (2.26) is attained since ω is continuous on X and X is compact, and all the minimizers belong to X^o, whence the minimizer is unique, since $V(x,\cdot)$ is strongly convex on X^o, and hence the prox-mapping is well defined. Using the definition of the prox-mapping for $\omega(x) = \frac{1}{2}\|x\|_2^2$, the iterative formula (2.1) can be written as

$$x_{j+1} = P_{x_j}(\gamma_j \mathsf{G}(x_j,\xi_j)), \quad x_1 \in X^o. \tag{2.27}$$

We discuss now the recursion (2.27) for general distance generating function $\omega(x)$. As it was mentioned above, if $\omega(x) = \frac{1}{2}\|x\|_2^2$, then formula (2.27) coincides with (2.1). In that case we refer to the procedure as *Euclidean* SA algorithm.

The following statement is a simple consequence of the optimality conditions of the right hand side of (2.26).

Lemma 2.1 *For any $u \in X, x \in X^o$ and y the following inequality holds*

$$V(P_x(y),u) \leq V(x,u) + y^T(u-x) + \frac{\|y\|_*^2}{2\alpha}. \tag{2.28}$$

Proof of this lemma is given in the Appendix.

Using (2.28) with $x = x_j$, $y = \gamma_j\mathsf{G}(x_j,\xi_j)$ and $u = \bar{x}$, we get

$$\gamma_j(x_j-\bar{x})^T\mathsf{G}(x_j,\xi_j) \leq V(x_j,\bar{x}) - V(x_{j+1},\bar{x}) + \frac{\gamma_j^2}{2\alpha}\|\mathsf{G}(x_j,\xi_j)\|_*^2. \tag{2.29}$$

If we compare inequality (2.29) with (2.2) we see that values of the prox-function $V(x_j, \bar{x})$ along the iterations of the Mirror Descent SA satisfy exactly the same relations as values $A_j = \frac{1}{2}\|x_j - \bar{x}\|_2^2$ along the trajectory of the Euclidean SA. The proposed construction of the prox-function $V(\cdot,\cdot)$ allows us to act in the general case exactly in the same way as we have done in the Euclidean situation of the previous section. Setting

$$\Delta_j := \mathsf{G}(x_j, \xi_j) - \mathsf{g}(x_j), \tag{2.30}$$

we can rewrite (2.29), with j replaced by t, as

$$\gamma_t (x_t - \bar{x})^T \mathsf{g}(x_t) \le V(x_t, \bar{x}) - V(x_{t+1}, \bar{x}) - \gamma_t \Delta_t^T (x_t - \bar{x}) + \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2. \tag{2.31}$$

Summing up over $t = 1, ..., j$, and taking into account that $V(x_{j+1}, u) \ge 0$, $u \in X$, we get

$$\sum_{t=1}^{j} \gamma_t (x_t - \bar{x})^T \mathsf{g}(x_t) \le V(x_1, \bar{x}) + \sum_{t=1}^{j} \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2 - \sum_{t=1}^{j} \gamma_t \Delta_t^T (x_t - \bar{x}). \tag{2.32}$$

Now let $\nu_t := \frac{\gamma_t}{\sum_{i=1}^{j} \gamma_i}$, $t = 1, ..., j$, and

$$\tilde{x}_j := \sum_{t=1}^{j} \nu_t x_t. \tag{2.33}$$

By convexity of $f(\cdot)$ we have that

$$\begin{array}{rcl} \sum_{t=1}^{j} \gamma_t (x_t - \bar{x})^T \mathsf{g}(x_t) & \ge & \sum_{t=1}^{j} \gamma_t \left[f(x_t) - f(\bar{x}) \right] = \left(\sum_{t=1}^{j} \gamma_t \right) \left[\sum_{t=1}^{j} \nu_t f(x_t) - f(\bar{x}) \right] \\ & \ge & \left(\sum_{t=1}^{j} \gamma_t \right) \left[f(\tilde{x}_j) - f(\bar{x}) \right]. \end{array}$$

Together with (2.32) this implies that

$$f(\tilde{x}_j) - f(\bar{x}) \le \frac{V(x_1, \bar{x}) + \sum_{t=1}^{j} \frac{\gamma_t^2}{2\alpha} \|\mathsf{G}(x_t, \xi_t)\|_*^2 - \sum_{t=1}^{j} \gamma_t \Delta_t^T (x_t - \bar{x})}{\sum_{t=1}^{j} \gamma_t}. \tag{2.34}$$

Let us suppose, as in the previous section (cf., (2.5)), that there is a constant $M_*^2 > 0$ such that

$$\mathbb{E}\left[\|\mathsf{G}(x, \xi)\|_*^2 \right] \le M_*^2, \ \ \forall x \in X. \tag{2.35}$$

Taking expectations of both sides of (2.34) and noting that: (i) x_t is a deterministic function of $\xi_{[t-1]} = (\xi_1, ..., \xi_{t-1})$, (ii) conditional on $\xi_{[t-1]}$ the expectation of Δ_t is 0, and (iii) the expectation of $\|\mathsf{G}(x_t, \xi_t)\|_*^2$ does not exceed M_*^2, we obtain

$$\mathbb{E}\left[f(\tilde{x}_j) - f(\bar{x}) \right] \le \frac{h_1 + (2\alpha)^{-1} M_*^2 \sum_{t=1}^{j} \gamma_t^2}{\sum_{t=1}^{j} \gamma_t}, \tag{2.36}$$

where $h_1 := \max_{u \in X} V(x_1, u)$.

To design the stepsize strategy let us start with the situation when the number j of iterations of the method is fixed in advance, say equals to N. Then the constant stepsize strategy $\gamma_t \equiv \gamma$,

$t = 1, ..., N$, can be implemented. Let us suppose from now on that the initial point x_1 is exactly the minimizer of $\omega(x)$ on X. Then $V(x_1, z) \leq D^2_{\omega,X}$, where

$$D_{\omega,X} := [\max_{z\in X} \omega(z) - \min_{z\in X} \omega(z)]^{1/2}, \tag{2.37}$$

and thus $h_1 \leq D^2_{\omega,X}$. Then the approximate solution $\tilde{x}_N$ satisfies:

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \leq \frac{D^2_{\omega,X}}{\gamma N} + \frac{M_*^2 \gamma}{2\alpha}, \tag{2.38}$$

where

$$\tilde{x}_N = \frac{1}{N}\sum_{t=1}^{N} x_t. \tag{2.39}$$

If we set

$$\gamma := \frac{D_{\omega,X}}{M_*}\sqrt{\frac{2\alpha}{N}}, \tag{2.40}$$

we get

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \leq D_{\omega,X} M_* \sqrt{\frac{2}{\alpha N}}. \tag{2.41}$$

We refer to the method (2.27), (2.33) and (2.40) as *Robust Mirror Descent SA* algorithm with constant stepsize policy.

By Markov inequality it follows from (2.41) that for any $\varepsilon > 0$,

$$\text{Prob}\left\{f(\tilde{x}_N) - f(\bar{x}) > \varepsilon\right\} \leq \frac{\sqrt{2} D_{\omega,X} M_*}{\varepsilon\sqrt{\alpha N}}. \tag{2.42}$$

It is possible, however, to obtain much finer bounds for those probabilities when imposing more restrictive assumptions on the distribution of $\mathsf{G}(x, \xi)$. Let us assume that

$$\mathbb{E}\left[\exp\left\{\|\mathsf{G}(x,\xi)\|_*^2 / M_*^2\right\}\right] \leq \exp\{1\}, \quad \forall x \in X. \tag{2.43}$$

Note that condition (2.43) is stronger than condition (2.35). Indeed, if a random variable Y satisfies $\mathbb{E}[\exp\{Y/a\}] \leq \exp\{1\}$ for some $a > 0$, then by Jensen inequality $\exp\{\mathbb{E}[Y/a]\} \leq \mathbb{E}[\exp\{Y/a\}] \leq \exp\{1\}$, and therefore $\mathbb{E}[Y] \leq a$. Of course, condition (2.43) holds if $\|\mathsf{G}(x, \xi)\|_* \leq M_*$ for all $(x, \xi) \in X \times \Xi$.

Proposition 2.1 *Suppose that condition* (2.43) *holds. Then for the constant stepsizes* (2.40) *the following inequality holds for any* $\Omega \geq 1$*:*

$$\text{Prob}\left\{f(\tilde{x}_N) - f(\bar{x}) > M_* D_{\omega,X}\sqrt{\frac{2}{\alpha N}}\,(12 + 2\Omega)\right\} \leq 2\exp\{-\Omega\}. \tag{2.44}$$

Proof of this proposition is given in the Appendix.

Discussion. The confidence bound (2.44) can be written as

$$\operatorname{Prob}\{f(\tilde{x}_N) - f(\bar{x}) > \varepsilon\} \le O(1)\exp\left\{-\kappa\varepsilon\sqrt{N}\right\}, \tag{2.45}$$

where $\kappa := \frac{\sqrt{\alpha}}{\sqrt{8}M_* D_{\omega,X}}$ and $\varepsilon > 0$. (Condition $\Omega \ge 1$ means here that $N \ge 49\kappa^{-2}\varepsilon^{-2}$.) It follows that for chosen accuracy ε and confidence level $\delta \in (0,1)$, the sample size

$$N \ge \frac{O(1)M_*^2 D_{\omega,X}^2 \ln^2(\delta^{-1})}{\alpha\varepsilon^2} \tag{2.46}$$

guarantees that $\tilde{x}_N$ is an ε-optimal solution of the true problem with probability at least $1-\delta$. This can be compared with a similar estimate for an optimal solution of the SAA problem (1.4) (cf., [16]). In both cases the estimated sample size N, considered as a function of the accuracy ε, is of order $O(\varepsilon^{-2})$ and depends logarithmically on the confidence level δ.

We can modify Robust Mirror Descent SA algorithm so that the approximate solution $\tilde{x}_N$ is obtained by averaging over a part of trajectory, namely, let for some integer ℓ, $1 \le \ell \le N$,

$$\tilde{x}_N := \frac{1}{\lfloor N/\ell \rfloor} \sum_{t=N-\lfloor N/\ell \rfloor+1}^{N} x_t.$$

In this case we have for the constant stepsize strategy with $\gamma := \frac{\overline{D}_{\omega,X}}{M_*}$,

$$\mathbb{E}\left[f(\tilde{x}_N) - f(\bar{x})\right] \le \frac{\overline{D}_{\omega,X} M_*(\ell+1)}{2\sqrt{\alpha N}},$$

where the quantity

$$\overline{D}_{\omega,X} := \left[2\sup\nolimits_{x\in X^o, z\in X} V(x,z)\right]^{1/2}$$

is assumed to be finite (which definitely is the case when ω is continuously differentiable on the entire X). Note that in the case of Euclidean SA, when $\omega(x) = \frac{1}{2}\|x\|_2^2$, $\overline{D}_{\omega,X}$ coincides with the Euclidean diameter $\overline{D}_X$ of X.

A decreasing stepsize strategy with

$$\gamma_t := \frac{\theta}{\sqrt{t}}, \quad t = 1,2,..., \tag{2.47}$$

can be also used in the Robust Mirror Descent algorithm. One can easily verify that when $\theta := \overline{D}_{\omega,X}/M_*$, the approximate solution $\tilde{x}_j$,

$$\tilde{x}_j = \left(\sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t\right)^{-1} \sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t x_t,$$

satisfies for $j \ge 2$ and $2 \le \ell \le j$:

$$\left[f(\tilde{x}_j) - f(\bar{x})\right] \le O(1)\frac{\ell\overline{D}_{\omega,X} M_*}{\sqrt{\alpha j}}. \tag{2.48}$$

We see that for both methods, (Euclidean) Robust SA and Robust Mirror Descent SA, the expected value of the error of the last iterate after t steps is of order $O(t^{-1/2})$. A potential benefit of the Mirror Descent over the Euclidean algorithm is that the norm $\|\cdot\|$ and the distance generating function $\omega(\cdot)$ can be adjusted to the geometry of the set X.

Example 2.1 Let $X := \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x \geq 0\}$ be a standard simplex. Suppose that the initial solution is the barycenter $x_1 = n^{-1}(1,1,...,1)$ of the simplex. In that case it is not difficult to find the exact Euclidean projection $\Pi_X(x)$. The estimate (2.19) suggests an error of order $D_X M N^{-1/2}$ of obtained solution for a sample of size N, with the constant M defined in (2.5). Here the (Euclidean) characteristics D_X of the set X, $D_X = \max_{x\in X}\|x - x_1\|_2 \leq \sqrt{2}$ for any $n \geq 1$.

Now let us consider the ℓ_1 norm $\|x\|_1 = \sum_{i=1}^n |x_i|$. Its dual norm is the ℓ_∞ norm $\|x\|_\infty = \max\{|x_1|, ..., |x_n|\}$. For $\omega(x) = \frac{1}{2}\|x\|_2^2$ the corresponding estimate (2.41) suggests an error of order $\alpha^{-1/2} D_{\omega,X} M_* N^{-1/2}$, where the constants α and M_* are computed with respect to the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$, respectively. We have that for any $x \in \mathbb{R}^n$,

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty \text{ and } \|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2,$$

and these inequalities are sharp. This indicates that the constant M_* might be up to $\sqrt{n}$-times smaller than the constant M. However, the constant α, taken with respect to the ℓ_1 norm is also $\sqrt{n}$-times smaller than the corresponding constant of ω taken with respect to the ℓ_2 norm. In other words, we do not gain anything in terms of the estimate (2.41) as compared with the estimate (2.23). This, of course, should be not surprising since the algorithm depends on a chosen norm only through the choice (2.40) of stepsizes.

Consider now the entropy distance generating function

$$\omega(x) := \sum_{i=1}^n x_i \ln x_i. \tag{2.49}$$

Here $X^o = \{x \in X : x > 0\}$, $D_{\omega,X} = \sqrt{\ln n}$, $x_1 := \text{argmin}_X \omega = n^{-1}(1, ..., 1)^T$ is the barycenter of X, and $\alpha = 1$ (see the Appendix). The corresponding prox-function $V(x,z)$ is

$$V(x,z) = \sum_{i=1}^n z_i \ln \frac{z_i}{x_i}.$$

Note that we can easily compute the prox mapping $P_x(y)$ of (2.26) in this case:

$$[P_x(y)]_i = \frac{x_i e^{-y_i}}{\sum_{k=1}^n x_k e^{-y_k}},\ \ i = 1, ..., n.$$

We can compare the Mirror Descent SA algorithm associated with the above choice of "entropy like" distance generating function coupled with the ℓ_1 norm and its dual ℓ_∞ norm, with the Euclidean SA algorithm. The error estimate (2.41) suggests that we lose a factor $\ln n$ in the ratio $\frac{D^2_{\omega,X}}{\alpha}$ as compared with $\frac{D_X^2}{\alpha}$. On the other hand, we have a potential gain of factor of order $\sqrt{n}$ in M_* (which is computed with respect to the norm ℓ_∞) as compared with M (computed with respect to the Euclidean norm).

3 Stochastic saddle point problem

We show in this section how the Mirror Descent SA algorithm can be modified to solve a convex-concave stochastic saddle point problem. Consider the following minimax (saddle point) problem

$$\min_{x\in X}\max_{y\in Y}\left\{\phi(x,y):=\mathbb{E}[\Phi(x,y,\xi)]\right\}. \tag{3.1}$$

Here $X\subset\mathbb{R}^n$ and $Y\subset\mathbb{R}^m$ are nonempty bounded closed convex sets, ξ is a random vector whose probability distribution P is supported on set $\Xi\subset\mathbb{R}^d$ and $\Phi: X\times Y\times\Xi\to\mathbb{R}$. We assume that for every $\xi\in\Xi$, function $\Phi(x,y,\xi)$ is convex in $x\in X$ and concave in $y\in Y$, and for all $x\in X$, $y\in Y$ the expectation

$$\mathbb{E}[\Phi(x,y,\xi)]=\int_\Xi \Phi(x,y,\xi)dP(\xi)$$

is well defined and finite valued. It follows that $\phi(x,y)$ is convex in $x\in X$ and concave in $y\in Y$, finite valued, and hence (3.1) is a *convex-concave saddle point problem.* In addition, we assume that $\phi(\cdot,\cdot)$ is *Lipschitz continuous* on $X\times Y$. It is well known that in the above setting the problem (3.1) is solvable, i.e., the corresponding "primal" and "dual" optimization problems $\min_{x\in X}\left[\max_{y\in Y}\phi(x,y)\right]$ and $\max_{y\in Y}\left[\min_{x\in X}\phi(x,y)\right]$, respectively, have optimal solutions and equal optimal values, denoted ϕ^*, and the pairs (x^*,y^*) of optimal solutions to the respective problems form the set of saddle points of $\phi(x,y)$ on $X\times Y$.

As in the case of the minimization problem (1.1) we assume that neither the function $\phi(x,y)$ nor its sub/supergradients in x and y are available explicitly. However, we make the following assumption.

(A2′) We have at our disposal an oracle which for every given $x\in X$, $y\in Y$ and $\xi\in\Xi$ returns value $\Phi(x,y,\xi)$ and a *stochastic subgradient*, that is, $(n+m)$-dimensional vector $\mathsf{G}(x,y,\xi)=\begin{bmatrix}\mathsf{G}_x(x,y,\xi)\\ -\mathsf{G}_y(x,y,\xi)\end{bmatrix}$ such that vector $\mathsf{g}(x,y)=\begin{bmatrix}\mathsf{g}_x(x,y)\\ -\mathsf{g}_y(x,y)\end{bmatrix}:=\begin{bmatrix}\mathbb{E}[\mathsf{G}_x(x,y,\xi)]\\ -\mathbb{E}[\mathsf{G}_y(x,y,\xi)]\end{bmatrix}$ is well defined, and $\mathsf{g}_x(x,y)\in\partial_x\phi(x,y)$ and $-\mathsf{g}_y(x,y)\in\partial_y(-\phi(x,y))$. For example, under mild assumptions we can set

$$\mathsf{G}(x,y,\xi)=\begin{bmatrix}\mathsf{G}_x(x,y,\xi)\\ -\mathsf{G}_y(x,y,\xi)\end{bmatrix}\in\begin{bmatrix}\partial_x\Phi(x,y,\xi)\\ \partial_y(-\Phi(x,y,\xi))\end{bmatrix}.$$

Let $\|\cdot\|_x$ be a norm on $\mathbb{R}^n$ and $\|\cdot\|_y$ be a norm on $\mathbb{R}^m$, and let $\|\cdot\|_{*,x}$ and $\|\cdot\|_{*,y}$ stand for the corresponding dual norms. As in Section 2.1, the basic assumption we make about the stochastic oracle (aside of its unbiasedness which we have already postulated) is that there exist positive constants $M^2_{*,x}$ and $M^2_{*,y}$ such that

$$\mathbb{E}\left[\|\mathsf{G}_x(u,v,\xi)\|^2_{*,x}\right]\le M^2_{*,x}\ \text{ and }\ \mathbb{E}\left[\|\mathsf{G}_y(u,v,\xi)\|^2_{*,y}\right]\le M^2_{*,y},\ \ \forall(u,v)\in X\times Y. \tag{3.2}$$

3.1 Mirror SA algorithm for saddle point problems

We equip X and Y with distance generating functions $\omega_x: X\to\mathbb{R}$ modulus α_x with respect to $\|\cdot\|_x$, and $\omega_y: Y\to\mathbb{R}$ modulus α_y with respect to $\|\cdot\|_y$. Let $D_{\omega_x,X}$ and $D_{\omega_y,Y}$ be the respective constants (see definition (2.37)). We equip $\mathbb{R}^n \times \mathbb{R}^m$ with the norm

$$\|(x,y)\| := \sqrt{\frac{\alpha_x}{2D^2_{\omega_x,X}}\|x\|_x^2 + \frac{\alpha_y}{2D^2_{\omega_y,Y}}\|y\|_y^2}, \tag{3.3}$$

so that the dual norm is

$$\|(\zeta,\eta)\|_* = \sqrt{\frac{2D^2_{\omega_x,X}}{\alpha_x}\|\zeta\|_{*,x}^2 + \frac{2D^2_{\omega_y,Y}}{\alpha_y}\|\eta\|_{*,y}^2}. \tag{3.4}$$

It follows by (3.2) that

$$\mathbb{E}\big[\|\mathsf{G}(x,y,\xi)\|_*^2\big] \le \frac{2D^2_{\omega_x,X}}{\alpha_x}M_{*,x}^2 + \frac{2D^2_{\omega_y,Y}}{\alpha_y}M_{*,y}^2 =: M_*^2. \tag{3.5}$$

We use notation $z = (x,y)$ and equip $Z := X \times Y$ with the distance generating function as follows:

$$\omega(z) := \frac{\omega_x(x)}{2D^2_{\omega_x,X}} + \frac{\omega_y(y)}{2D^2_{\omega_y,Y}}.$$

It is immediately seen that ω indeed is a distance generating function for Z modulus $\alpha = 1$ with respect to the norm $\|\cdot\|$, and that $Z^o = X^o \times Y^o$ and $D_{\omega,Z} = 1$. In what follows, $V(z,u) : Z^o \times Z \to \mathbb{R}$ and $P_z(\zeta) : \mathbb{R}^{n+m} \to Z^o$ are the prox-function and prox-mapping associated with ω and Z, see (2.25), (2.26).

We are ready now to present the Mirror SA algorithm for saddle point problems. This is the iterative procedure

$$z_{j+1} := P_{z_j}(\mathsf{G}(z_j,\xi_j)), \tag{3.6}$$

where the initial point $z_1 \in Z$ is chosen to be the minimizer of $\omega(z)$ on Z. As before (cf., (2.39)), we define the approximate solution $\tilde{z}_j$ of (3.1) after j iterations as

$$\tilde{z}_j = (\tilde{x}_j, \tilde{y}_j) := \left(\sum_{t=1}^{j}\gamma_t\right)^{-1}\sum_{t=1}^{j}\gamma_t z_t. \tag{3.7}$$

We refer to the procedure (3.6), (3.7) as *Saddle Point Mirror SA* algorithm.

Let us analyze the convergence properties of the algorithm. We measure quality of an approximate solution $\tilde{z} = (\tilde{x}, \tilde{y})$ by the error

$$\epsilon_\phi(\tilde{z}) := \left[\max_{y\in Y}\phi(\tilde{x},y) - \phi_*\right] + \left[\phi_* - \min_{x\in X}\phi(x,\tilde{y})\right] = \max_{y\in Y}\phi(\tilde{x},y) - \min_{x\in X}\phi(x,\tilde{y}).$$

By convexity of $\phi(\cdot,y)$ we have

$$\phi(x_t,y_t) - \phi(x,y_t) \le \mathsf{g}_x(x_t,y_t)^T(x_t - x), \quad \forall x \in X,$$

and by concavity of $\phi(x,\cdot)$,

$$\phi(x_t,y) - \phi(x_t,y_t) \le \mathsf{g}_y(x_t,y_t)^T(y - y_t), \quad \forall y \in Y,$$

so that for all $z = (x, y) \in Z$,

$$\phi(x_t, y) - \phi(x, y_t) \le \mathsf{g}_x(x_t, y_t)^T(x_t - x) + \mathsf{g}_y(x_t, y_t)^T(y - y_t) = \mathsf{g}(z_t)^T(z_t - z).$$

Using once again the convexity-concavity of ϕ we write

$$\begin{aligned}
\epsilon_\phi(\tilde{z}_j) &= \max_{y\in Y} \phi(\tilde{x}_j, y) - \min_{x\in X} \phi(x, \tilde{y}_j) \\
&\le \left[\sum_{t=1}^{j} \gamma_t\right]^{-1} \left[\max_{y\in Y} \sum_{t=1}^{j} \gamma_t \phi(x_t, y) - \min_{x\in X} \sum_{t=1}^{j} \gamma_t \phi(x, y_t)\right] \\
&\le \left(\sum_{t=1}^{j} \gamma_t\right)^{-1} \max_{z\in Z} \sum_{t=1}^{j} \gamma_t \mathsf{g}(z_t)^T(z_t - z). \qquad (3.8)
\end{aligned}$$

To bound the right-hand side of (3.8) we use the following result.

Lemma 3.1 *In the above setting, for any* $j \ge 1$ *the following inequality holds*

$$\mathbb{E}\left[\max_{z\in Z} \sum_{t=1}^{j} \gamma_t \mathsf{g}(z_t)^T(z_t - z)\right] \le 2 + \frac{5}{2} M_*^2 \sum_{t=1}^{j} \gamma_t^2. \qquad (3.9)$$

Proof of this lemma is given in the Appendix.

Now to get an error bound for the solution $\tilde{z}_j$ it suffices to substitute inequality (3.9) into (3.8) to obtain

$$\mathbb{E}[\epsilon_\phi(\tilde{z}_j)] \;\le\; \left(\sum_{t=1}^{j} \gamma_t\right)^{-1} \left[2 + \frac{5}{2} M_*^2 \sum_{t=1}^{j} \gamma_t^2\right].$$

Let us use the constant stepsize strategy

$$\gamma_t = \frac{2}{M_*\sqrt{5N}}, \quad t = 1, ..., N. \qquad (3.10)$$

Then $\epsilon_\phi(\tilde{z}_N) \le 2M_*\sqrt{\frac{5}{N}}$, and hence (see definition (3.5) of M_*) we obtain

$$\epsilon_\phi(\tilde{z}_N) \le 2\sqrt{\frac{10\left[\alpha_y D^2_{\omega_x,X} M^2_{*,x} + \alpha_x D^2_{\omega_y,Y} M^2_{*,y}\right]}{\alpha_x \alpha_y N}}. \qquad (3.11)$$

Same as in the minimization case, we can pass from constant stepsizes on a fixed "time horizon" to decreasing stepsize policy (2.47) with $\theta = 1/M_*$ and from averaging of all iterates to the "sliding averaging"

$$\tilde{z}_j = \left(\sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t\right)^{-1} \sum_{t=j-\lfloor j/\ell \rfloor}^{j} \gamma_t z_t,$$

arriving at the efficiency estimate

$$\epsilon(\tilde{z}_j) \le O(1) \frac{\ell \overline{D}_{\omega,Z} M_*}{\sqrt{j}}, \qquad (3.12)$$

where the quantity $\overline{D}_{\omega,Z} = \left[2\sup_{z\in Z^o, w\in Z} V(z,w)\right]^{1/2}$ is assumed to be finite.

We give below a bound on the probabilities of large deviations of the error $\epsilon_\phi(\tilde{z}_N)$.

Proposition 3.1 *Suppose that conditions of the bound* (3.11) *are verified and, further, it holds for all* $(u,v) \in Z$ *that*

$$\mathbb{E}\left[\exp\left\{\|\mathsf{G}_x(u,v,\xi)\|_{*,x}^2 / M_{*,x}^2\right\}\right] \le \exp\{1\}, \quad \mathbb{E}\left[\exp\left\{\|\mathsf{G}_y(x,y,\xi)\|_{*,y}^2 / M_{*,y}^2\right\}\right] \le \exp\{1\}. \tag{3.13}$$

Then for the stepsizes (3.10) *one has for any* $\Omega \ge 1$ *that*

$$\mathrm{Prob}\left\{\epsilon_\phi(\tilde{z}_N) > \frac{(8+2\Omega)\sqrt{5}M_*}{\sqrt{N}}\right\} \le 2\exp\{-\Omega\}. \tag{3.14}$$

Proof of this proposition is given in the Appendix.

3.2 Application to minimax stochastic problems

Consider the following minimax stochastic problem

$$\min_{x\in X} \max_{1\le i\le m} \left\{f_i(x) := \mathbb{E}[F_i(x,\xi)]\right\}, \tag{3.15}$$

where $X \subset \mathbb{R}^n$ is a nonempty bounded closed convex set, ξ is a random vector whose probability distribution P is supported on set $\Xi \subset \mathbb{R}^d$ and $F_i : X \times \Xi \to \mathbb{R}$, $i = 1,...,m$. We assume that for a.e. ξ the functions $F_i(\cdot,\xi)$ are convex and for every $x \in \mathbb{R}^n$, $F_i(x,\cdot)$ are integrable, i.e., the expectations

$$\mathbb{E}[F_i(x,\xi)] = \int_\Xi F_i(x,\xi)dP(\xi), \quad i = 1,...,m, \tag{3.16}$$

are well defined and finite valued. To find a solution to the minimax problem (3.15) is exactly the same as to solve the saddle point problem

$$\min_{x\in X}\max_{y\in Y}\left\{\phi(x,y) := \sum_{i=1}^m y_i f_i(x)\right\}, \tag{3.17}$$

with $Y := \{y \in \mathbb{R}^m : \, y \ge 0, \, \sum_{i=1}^m y_i = 1\}$.

Similarly to assumptions (A1) and (A2), assume that we cannot compute $f_i(x)$ (and thus $\phi(x,y)$) explicitly, but are able to generate independent realizations $\xi_1, \xi_2, ...$ distributed according to P, and for given $x \in X$ and $\xi \in \Xi$ we can compute $F_i(x,\xi)$ and its *stochastic subgradient* $\mathsf{G}_i(x,\xi)$, i.e., such that $\mathsf{g}_i(x) = \mathbb{E}[\mathsf{G}_i(x,\xi)]$ is well defined and $\mathsf{g}_i(x) \in \partial f_i(x)$, $x \in X$, $i = 1,...,m$. In other words we have a stochastic oracle for the problem (3.17) such that assumption (A2′) holds, with

$$\mathsf{G}(x,y,\xi) := \begin{bmatrix} \sum_{i=1}^m y_i \mathsf{G}_i(x,\xi) \\ \big(-F_1(x,\xi), ..., -F_m(x,\xi)\big) \end{bmatrix}, \tag{3.18}$$

and

$$\mathsf{g}(x,y) := \mathbb{E}[\mathsf{G}(x,y,\xi)] = \begin{bmatrix} \sum_{i=1}^m y_i \mathsf{g}_i(x) \\ (-f_1(x), ..., -f_m(x)) \end{bmatrix} \in \begin{bmatrix} \partial_x \phi(x,y) \\ -\partial_y \phi(x,y) \end{bmatrix}. \tag{3.19}$$

Suppose that the set X is equipped with norm $\|\cdot\|_x$, whose dual norm is $\|\cdot\|_{*,x}$, and a distance generating function ω modulus α_x with respect to $\|\cdot\|_x$, and let $R_x^2 := \frac{D^2_{\omega_x,X}}{\alpha_x}$. We equip the set Y with norm $\|\cdot\|_y := \|\cdot\|_1$, so that $\|\cdot\|_{*,y} = \|\cdot\|_\infty$, and with the distance generating function

$$\omega_y(y) := \sum_{i=1}^m y_i \ln y_i,$$

and set $R_y^2 := \frac{D^2_{\omega_y,Y}}{\alpha_y} = \ln m$. Next, following (3.3) we set

$$\|(x,y)\| := \sqrt{\frac{\|x\|_x^2}{2R_x^2} + \frac{\|y\|_1^2}{2R_y^2}},$$

and hence

$$\|(\zeta,\eta)\|_* = \sqrt{2R_x^2\|\zeta\|_{*,x}^2 + 2R_y^2\|\eta\|_\infty^2}.$$

Let us assume uniform bounds:

$$\mathbb{E}\left[\max_{1\le i\le m}\|\mathsf{G}_i(x,\xi)\|_{*,x}^2\right] \le M_{*,x}^2, \quad \mathbb{E}\left[\max_{1\le i\le m}|F_i(x,\xi)|^2\right] \le M_{*,y}^2, \quad i=1,...,m.$$

Note that

$$\begin{aligned} \mathbb{E}\left[\|\mathsf{G}(x,y,\xi)\|_*^2\right] &= 2R_x^2\,\mathbb{E}\left[\|\sum_{i=1}^m y_i\mathsf{G}_i(x,\xi)\|_{*,x}^2\right] + 2R_y^2\,\mathbb{E}\left[\|F(x,\xi)\|_\infty^2\right] &(3.20)\\ &\le 2R_x^2M_{*,x}^2 + 2R_y^2M_{*,y}^2 = 2R_x^2M_{*,x}^2 + 2M_{*,y}^2\ln m =: M_*^2. &(3.21)\end{aligned}$$

Let us now use the Saddle Point Mirror SA algorithm (3.6), (3.7) with the constant stepsize strategy

$$\gamma_t = \frac{2}{M_*\sqrt{5N}}, \quad t=1,2,...,N.$$

When substituting the value of M_*, we obtain from (3.11):

$$\begin{aligned} \mathbb{E}\left[\epsilon_\phi(\tilde{z}_N)\right] = \mathbb{E}\left[\max_{y\in Y}\phi(\hat{x}_N,y) - \min_{x\in X}\phi(x,\hat{y}_N)\right] \le 2M_*\sqrt{\frac{5}{N}}\\ \le\ 2\sqrt{\frac{10\left[R_x^2M_{*,x}^2 + M_{*,x}^2\ln m\right]}{N}}. \qquad (3.22)\end{aligned}$$

Discussion. Looking at the bound (3.22) one can make the following important observation. The error of the Saddle Point Mirror SA algorithm in this case is “almost independent” of the number m of constraints (it grows as $O(\sqrt{\ln m})$ as m increases). The interested reader can easily verify that if an Euclidean SA algorithm were used in the same setting (i.e., the algorithm tuned to the norm $\|\cdot\|_y := \|\cdot\|_2$), the corresponding bound would grow with m much faster (in fact, our error bound would be $O(\sqrt{m})$ in that case).

Note that properties of the Saddle Point Mirror SA can be used to reduce significantly the arithmetic cost of the algorithm implementation. To this end let us look at the definition (3.18) of the stochastic oracle: in order to obtain a realization $\mathsf{G}(x,y,\xi)$ one has to compute m random subgradients $\mathsf{G}_i(x,\xi)$, $i=1,...,m$, and then the convex combination $\sum_{i=1}^m y_i \mathsf{G}_i(x,\xi)$. Now let η be an independent of ξ and uniformly distributed in $[0,1]$ random variable, and let $\imath(\eta,y):[0,1]\times Y\to\{1,...,m\}$ equals to i when $\sum_{s=1}^{i-1} y_s < \eta \le \sum_{s=1}^{i} y_s$. That is, random variable $\hat{\imath}=\imath(\eta,y)$ takes values $1,...,m$ with probabilities $y_1,...,y_m$. Consider random vector

$$\mathsf{G}(x,y,(\xi,\eta)) := \left[\begin{array}{c} \mathsf{G}_{\imath(\eta,y)}(x,\xi) \\ (-F_1(x,\xi),...,-F_m(x,\xi)) \end{array}\right]. \tag{3.23}$$

We refer to $\mathsf{G}(x,y,(\xi,\eta))$ as a *randomized oracle* for problem (3.17), the corresponding random parameter being (ξ,η). By construction we still have $\mathbb{E}\big[\mathsf{G}(x,y,(\xi,\eta))\big]=\mathsf{g}(x,y)$, where g is defined in (3.19), and, moreover, the same bound (3.20) holds for $\mathbb{E}\big[\|\mathsf{G}(x,y,(\xi,\eta))\|_*^2\big]$. We conclude that the accuracy bound (3.22) holds for the error of the Saddle Point Mirror SA algorithm with randomized oracle. On the other hand, in the latter procedure only one randomized subgradient $\mathsf{G}_{\hat{\imath}}(x,\xi)$ per iteration is to be computed. This simple idea is further developed in another interesting application of the Saddle Point Mirror SA algorithm to bilinear matrix games which we discuss next.

3.3 Application to bilinear matrix games

Consider the standard matrix game problem, that is, problem (3.1) with

$$\phi(x,y) := y^TAx + b^Tx + c^Ty,$$

where $A\in\mathbb{R}^{m\times n}$, and X and Y are the standard simplices, i.e.,

$$X := \left\{x\in\mathbb{R}^n : x\ge 0, \textstyle\sum_{j=1}^n x_j = 1\right\},\ \ Y := \left\{y\in\mathbb{R}^m : y\ge 0, \textstyle\sum_{i=1}^m y_i = 1\right\}.$$

In the case in question it is natural to equip X (respectively, Y) with the usual $\|\cdot\|_1$-norm on $\mathbb{R}^n$ (respectively, $\mathbb{R}^m$). We choose entropies as the corresponding distance generating functions:

$$\omega_x(x) := \sum_{i=1}^n x_i \ln x_i, \quad \omega_y(x) := \sum_{i=1}^m y_i \ln y_i.$$

As we already have seen, this choice results in $\frac{D^2_{\omega_x,X}}{\alpha_x}=\ln n$ and $\frac{D^2_{\omega_y,Y}}{\alpha_y}=\ln m$. According to (3.3) we set

$$\|(x,y)\| := \sqrt{\frac{\|x\|_1^2}{2\ln n}+\frac{\|y\|_1^2}{2\ln m}},$$

and thus

$$\|(\zeta,\eta)\|_* = \sqrt{2\|\zeta\|_\infty^2 \ln n + 2\|\eta\|_\infty^2 \ln m}. \tag{3.24}$$

In order to compute the estimates $\Phi(x,y,\xi)$ of $\phi(x,y)$ and $\mathsf{G}(x,y,\xi)$ of $\mathsf{g}(x,y)=(b+A^Ty,-c-Ax)$ to be used in the Saddle Point Mirror SA iterations (3.6), we use the *randomized oracle*

$$\begin{array}{rcl} \Phi(x,y,\xi) & = & c^Tx + b^Ty + A_{\imath(\xi_1,y)\imath(\xi_2,x)}, \\ \mathsf{G}(x,y,\xi) & = & \left[\begin{array}{c} c + A^{\imath(\xi_1,y)} \\ -b - A_{\imath(\xi_2,x)} \end{array}\right], \end{array}$$

where ξ_1 and ξ_2 are independent uniformly distributed on $[0,1]$ random variables, $\hat{j} = \imath(\xi_1, y)$ and $\hat{i} = \imath(\xi_2, x)$ are defined as in (3.23), i.e., $\hat{j}$ can take values $1, ..., m$ with probabilities $y_1, ..., y_m$ and $\hat{i}$ can take values $1, ..., n$ with probabilities $x_1, ..., x_n$, and A_j, $[A^i]^T$ are j-th column and i-th row in A, respectively.

Note that $\mathsf{g}(x,y) := \mathbb{E}\big[\mathsf{G}(x,y,(\hat{j},\hat{i}))\big] \in \left[\begin{array}{c} \partial_x \phi(x,y) \\ \partial_y(-\phi(x,y)) \end{array}\right]$. Besides this,

$$|\mathsf{G}(x,y,\xi)_i| \leq \max_{1\leq j\leq m} \|A^j + b\|_\infty, \ \text{ for } i = 1, ..., n,$$

and

$$|\mathsf{G}(x,y,\xi)_i| \leq \max_{1\leq j\leq n} \|A_j + c\|_\infty, \ \text{ for } i = n+1, ..., n+m.$$

Hence, by the definition (3.24) of $\|\cdot\|_*$,

$$\mathbb{E}\|\mathsf{G}(x,y,\xi)\|_*^2 \ \leq \ M_*^2 := 2\ln n \max_{1\leq j\leq m} \|A^j + b\|_\infty^2 + 2\ln m \max_{1\leq j\leq n} \|A_j + c\|_\infty^2.$$

The bottom line is that inputs of the randomized Mirror Saddle Point SA satisfy the conditions of validity of the bound (3.11) with M_* as above. Using the constant stepsize strategy with

$$\gamma_t = \frac{2}{M_*\sqrt{5N}}, \quad t = 1, ..., N,$$

we obtain from (3.11):

$$\mathbb{E}\big[\epsilon_\phi(\tilde{z}_N)\big] = \mathbb{E}\left[\max_{y\in Y} \phi(\tilde{x}_N, y) - \min_{x\in X} \phi(x, \tilde{y}_N)\right] \leq 2M_*\sqrt{\frac{5}{N}}. \tag{3.25}$$

We continue with the counterpart of Proposition 3.1 for the Saddle Point Mirror SA in the setting of bilinear matrix games.

Proposition 3.2 *For any $\Omega \geq 1$ it holds that*

$$\text{Prob}\left\{\epsilon_\phi(\tilde{z}_N) > 2M_*\sqrt{\frac{5}{N}} + \frac{4\overline{M}}{\sqrt{N}}\Omega\right\} \leq \exp\left\{-\Omega^2/2\right\}, \tag{3.26}$$

where

$$\overline{M} := \max_{1\leq j\leq m} \|A^j + b\|_\infty + \max_{1\leq j\leq n} \|A_j + c\|_\infty. \tag{3.27}$$

Discussion. Consider a bilinear matrix game with $m = n$ and $b = c = 0$. Suppose that we are interested to solve it within a fixed relative accuracy ρ, that is, to ensure that a (perhaps random) approximate solution $\tilde{z}_N$, we get after N iterations, satisfies the error bound

$$\epsilon_\phi(\tilde{z}_N) \leq \rho \max_{1\leq i,j\leq n} |A_{ij}|$$

with probability at least $1-\delta$. According to (3.26), to this end one can use the randomized Saddle Point Mirror SA algorithm (3.6), (3.7) with

$$N = O(1)\frac{\ln n + \ln(\delta^{-1})}{\rho^2}. \tag{3.28}$$

The computational cost of building $\tilde{z}_N$ with this approach is

$$\mathcal{C}(\rho) = O(1)\frac{\left[\ln n + \ln(\delta^{-1})\right]\mathcal{R}}{\rho^2}$$

arithmetic operations, where $\mathcal{R}$ is the arithmetic cost of extracting a column/row from A, given the index of this column/row. The total number of rows and columns visited by the algorithm does not exceed the sample size N, given in (3.28), so that the total number of entries in A used in course of the entire computation does not exceed

$$M = O(1)\frac{n(\ln n + \ln(\delta^{-1}))}{\rho^2}.$$

When ρ is fixed and n is large, this is incomparably less that the total number n^2 of entries of A. Thus, *the algorithm in question produces reliable solutions of prescribed quality to large-scale matrix games by inspecting a negligible, as $n \to \infty$, part of randomly selected data.* Note that randomization here is critical. It is easily seen that a deterministic algorithm which is capable to find a solution with (deterministic) relative accuracy $\rho \leq 0.1$, has to "see" in the worst case *at least* $O(1)n$ rows/columns of A.

4 Numerical results

In this section, we report the results of our computational experiments where we compare the performance of the Robust Mirror Descent SA method and the SAA method applied to three stochastic programming problems, namely: a stochastic utility problem, a stochastic max-flow problem and network planning problem with random demand. We also present a small simulation study of performance of randomized Mirror SA algorithm for bilinear matrix games.

4.1 A stochastic utility problem

Our first experiment was carried out with the utility model

$$\min_{x \in X} \left\{ f(x) := \mathbb{E}\left[\phi\left(\textstyle\sum_{i=1}^n (a_i + \xi_i) x_i\right)\right] \right\}. \tag{4.1}$$

Here $X = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x \geq 0\}$, $\xi_i \sim N(0,1)$ are independent random variables having standard normal distribution, $a_i = i/n$ are constants, and $\phi(\cdot)$ is a piecewise linear convex function given by $\phi(t) = \max\{v_1 + s_1 t, ..., v_m + s_m t\}$, where v_k and s_k are certain constants.

Two variants of the Robust Mirror Descent SA method have been used for solving problem (4.1). The first variant, the *Euclidean* SA (E-SA), employs the Euclidean distance generating function $\omega(x) = \frac{1}{2}\|x\|_2^2$, and its iterates coincide with those of the classic SA as discussed in Section 2.3. The other distance generating function used in the following experiments is the entropy function defined in (2.49). The resulting variant of the Robust Mirror Descent SA is referred to as the *Non-Euclidean* SA (N-SA) method.

These two variants of SA method are compared with the SAA approach in the following way: fixing an i.i.d. sample (of size N) for the random variable ξ, we apply the three afore-mentioned methods to obtain approximate solutions for problem (4.1), and then the quality of the solutions

Table 1: the selection of step-sizes
[method: N-SA, N:2,000, K:10,000, instance: L1]

	η			
policy	0.1	1	5	10
variable	-7.4733	-7.8865	-7.8789	-7.8547
constant	-6.9371	-7.8637	-7.9037	-7.8971

yielded by these algorithms is evaluated using another i.i.d. sample of size $K >> N$. It should be noted that SAA itself is not an algorithm and in our experiments it is coupled with the so-called Non-Euclidean Restricted Memory Level (NERML) deterministic algorithm (see [2]), for solving the sample average problem (1.4).

In our experiment, the function $\phi(\cdot)$ in problem (4.1) was set to be a piecewise linear function with 10 breakpoints ($m = 10$) over the interval $[0, 1]$ and four instances (namely: L1, L2, L3 and L4) with different dimensions ranging from $n = 500$ to $5,000$ were randomly generated. Note that each of these instances assumes a different function $\phi(\cdot)$, i.e., has different values of v_k and s_k for $k = 1, ..., m$. All the algorithms were coded in ANSI C and the experiments were conducted on a Intel PIV 1.6GHz machine with Microsoft Windows XP professional.

The first step of our experimentation is to determine the step-sizes γ_t used by both variants of SA method. Note that in our situation, either a constant stepsize policy (2.40) or a variable step-size policy (2.47) can be applied. Observe however that the quantity M_* appearing in both stepsize policies is usually unknown and requires an estimation. In our implementation, an estimate of M_* is obtained by taking the maxima of $\|\mathsf{G}(\cdot, \cdot)\|_*$ over a certain number (for example, 100) of random feasible solutions x and realizations of the random variable ξ. To account for the error inherited by this estimation procedure, the stepsizes are set to $\eta\gamma_t$ for $t = 1, ..., N$, where γ_t are defined as in (2.40) or (2.47), and $\eta > 0$ is a user-defined parameter that can be fine-tuned, for example, by a trial-and-error procedure.

Some results of our experiments for determining the step-sizes are presented in Table 1. Specifically, Table 1 compares the solution quality obtained by the *Non-Euclidean* SA method (N = 2,000 and K = 10,000) applied for solving the instance L1 ($n = 500$) with different stepsize polices and different values of η. In this table, column 1 gives the name of the two policies and column 2 through column 5 report the objective values for $\eta = 0.1, 1, 5$ and 10 respectively. The results given in Table 1 show that the constant step-size policy with a properly chosen parameter η slightly outperforms the variable stepsize policy and the same phenomenon has also been observed for the *Euclidean* SA method. Based on these observations, the constant step-size was chosen for both variants of SA method. To set the parameter η, we run each variant of SA method in which different values of $\eta \in \{0.1, 1, 5, 10\}$ are applied, and the best selection of η in terms of the solution quality was chosen. More specifically, the parameter η was set to 0.1 and 5, respectively, for the *Euclidean* SA method and *Non-Euclidean* throughout our computation.

We then run each of the three afore-mentioned methods with various sample-sizes for each test instance and the computational results are reported in Table 2, where n is the dimension of problem, N denotes the sample-size, 'OBJ' and 'DEV' represents mean and deviation, respectively, of the objective values of problem (4.1) as evaluated over a sample of size $K = 10,000$ for the solutions generated by the algorithms, 'TIME' is the CPU seconds for obtaining the solutions, and 'ORC' stands for the number of calls to the stochastic oracle.

Table 2: SA vs. SAA on the stochastic utility model

-		L1: $n = 500$				L2: $n = 1000$			
ALG.	N	OBJ	DEV	TIME	ORC	OBJ	DEV	TIME	ORC
N-SA	100	-7.7599	0.5615	0.00	200	-5.8340	0.1962	0.00	200
	1,000	-7.8781	0.3988	2.00	1,100	-5.9152	0.1663	2.00	1,100
	2,000	-7.8987	0.3589	2.00	2,100	-5.9243	0.1668	5.00	2,100
	4,000	-7.9075	0.3716	5.00	4,100	-5.9365	0.1627	12.00	4,100
E-SA	100	-7.6895	0.3702	0.00	200	-5.7988	0.1046	1.00	200
	1,000	-7.8559	0.3153	2.00	1,100	-5.8919	0.0998	4.00	1,100
	2,000	-7.8737	0.3101	3.00	2,100	-5.9067	0.1017	7.00	2,100
	4,000	-7.8948	0.3084	7.00	4,100	-5.9193	0.1060	13.00	4,100
SAA	100	-7.6571	0.9343	7.00	4,000	-5.6346	0.9333	8.00	4,000
	1,000	-7.8821	0.4015	31.00	40,000	-5.9221	0.2314	68.00	40000
	2,000	-7.9100	0.3545	72.00	80,000	-5.9313	0.2100	128.00	80,000
	4,000	-7.9087	0.3696	113.00	160,000	-5.9384	0.1944	253.00	160,000
-		L3: $n = 2000$				L4: $n = 5000$			
ALG.	N	OBJ	DEV	TIME	ORC	OBJ	DEV	TIME	ORC
N-SA	100	-7.1419	0.2394	1.00	200	-5.4688	0.2719	3.00	200
	1,000	-7.2312	0.1822	6.00	1,100	-5.5716	0.1762	13.00	1,100
	2,000	-7.2513	0.1691	10.00	2,100	-5.5847	0.1506	25.00	2,100
	4,000	-7.2595	0.1685	20.00	4,100	-5.5935	0.1498	49.00	4,100
E-SA	100	-7.0165	0.1547	1.00	200	-4.9364	0.1111	4.00	200
	1,000	-7.2029	0.1301	7.00	1,100	-5.3895	0.1416	20.00	1,100
	2,000	-7.2306	0.1256	15.00	2,100	-5.4870	0.1238	39.00	2,100
	4,000	-7.2441	0.1282	29.00	4,100	-5.5354	0.1195	77.00	4,100
SAA	100	-6.9748	0.8685	19.00	4,000	-5.3360	0.7188	44.00	4,000
	1,000	-7.2393	0.2469	134.00	40,000	-5.5656	0.2181	337.00	40,000
	2,000	-7.2583	0.2030	261.00	80,000	-5.5878	0.1747	656.00	80,000
	4,000	-7.2664	0.1838	515.00	160,000	-5.5967	0.1538	1283.00	160,000

In order to evaluate variability of these algorithms, we run each method 100 times and compute the resulting statistics as shown in Table 3. Note that the instance L2 is chosen as a representative and only two different sample-sizes ($N = 1000$ and 2000) are applied since this test is more time-consuming. In Table 3, column 1 and column 2 give the instance name and the sample-size used for each run of the method. The objective value of the approximate solution yielded by each run of the algorithm was evaluated over $K = 10^4$ sample size, and the mean and standard deviation of these objective values over 100 runs are given in columns 3-4, columns 6-7, and columns 9-10, respectively, for N-SA, E-SA and SAA method. The average solution time of these three methods over 100 runs are also reported in column 5, 8, and 11 respectively.

The experiment demonstrates that the solution quality is improved for all three methods with the increase of the sample size N. Moreover, for a given sample size, the solution time for N-SA is significantly smaller than that for SAA, while the solution quality for N-SA is close to that for the latter one.

Table 3: The variability for the stochastic utility problem

-		N-SA			E-SA			SAA		
		OBJ		AVG.	OBJ		AVG.	OBJ		AVG.
INST	N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
L2	1,000	-5.9159	0.0025	2.63	-5.8925	0.0024	4.99	-5.9219	0.0047	67.31
L2	2,000	-5.9258	0.0022	5.03	-5.9063	0.0019	7.09	-5.9328	0.0028	131.25

4.2 Stochastic max-flow problem

In the second experiment, we consider a simple two-stage stochastic linear programming, namely, a stochastic max-flow problem. The problem is to investigate the capacity expansion over a stochastic network. Let $G = (N, A)$ be a diagraph with a source node s and a sink node t. Each arc $(i, j) \in A$ has an existing capacity $p_{ij} \geq 0$, and a random implementing/operating level ξ_{ij}. Moreover, there is a common random degrading factor denoted by θ for all arcs in A. The goal is to determine how much capacity to add to the arcs subject to a budget constraint, such that the expected maximum flow from s to t is maximized. Let x_{ij} denote the capacity to be added to arc (i, j). The problem can be formulated as

$$\begin{array}{ll} \max\limits_{x} & \big\{f(x) := \mathbb{E}[F(x,\xi)]\big\} \\ \text{s.t.} & \sum\limits_{(i,j)\in A} c_{ij}x_{ij} \leq b, x_{ij} \geq 0, \quad \forall (i,j) \in A, \end{array} \tag{4.2}$$

where c_{ij} is unit cost for the capacity to be added, b is the total available budget, and $F(x, \xi)$ denotes the maximum $s - t$ flow in the network when the capacity of an arc (i, j) is given by $\theta\xi_{ij}(p_{ij} + x_{ij})$. Note that the above is a maximization rather than minimization problem.

For our purpose, we assume that the random variables ξ_{ij} and θ are independent and uniformly distributed over $(0, 1)$ and $(0.5, 1)$, respectively. Also let $p_{ij} = 0$ and $c_{ij} = 1$ for all $(i, j) \in E$, and $b = 1$. We randomly generated 4 network instances (referred to as F1, F2, F3 and F4) using the network generator GRIDGEN, which is available on DIMACS challenge. The push-relabel algorithm (see [5]) was used to solve the second stage max-flow problem.

The three methods, namely: N-SA, E-SA and SAA, and the same stepsize policy as discussed in Subsection 4.1, were applied for solving these stochastic max-flow instances. In the first test, each algorithm was run once for each test instance and the computational results are reported in Table 4, where m and n denote the number of nodes and arcs in G, respectively, N denotes the number of samples, 'OBJ' and 'DEV' represent the mean and standard deviation, respectively, of objective values of problem (4.2) as evaluated over $K = 10^4$ sample size at the approximated solutions yielded by the algorithms, 'TIME' is CPU seconds for obtaining the approximated solution, and 'ORC' stands for the number of calls to the stochastic oracle. Similar to the stochastic utility problem, we investigate the variability of these three methods by running each method for 100 times and computing the statistical results as shown in Table 5 whose columns have exactly the same meaning as in Table 3.

This experiment, once more, shows that for a given sample size N, the solution quality for N-SA is close to or even in some cases is better than that for SAA, meanwhile, the solution time of N-SA is much smaller than the latter one.

Table 4: SA vs. SAA on the stochastic max-flow model

-		F1: $m = 50, n = 500$				F2: $m = 100, n = 1000$			
ALG.	N	OBJ	DEV	TIME	ORC	OBJ	DEV	TIME	ORC
N-SA	100	0.1140	0.0786	0.00	200	0.0637	0.0302	0.00	200
	1000	0.1254	0.0943	1.00	1,100	0.0686	0.0300	3.00	1,100
	2000	0.1249	0.0947	3.00	2,100	0.0697	0.0289	6.00	2,100
	4000	0.1246	0.0930	5.00	4,100	0.0698	0.0268	11.00	4,100
E-SA	100	0.0840	0.0362	0.00	200	0.0618	0.0257	1.00	200
	1000	0.1253	0.0944	3.00	1,100	0.0670	0.0248	6.00	1,100
	2000	0.1246	0.0947	5.00	2,100	0.0695	0.0263	13.00	2,100
	4000	0.1247	0.0929	9.00	4,100	0.0696	0.0264	24.00	4,100
SAA	100	0.1212	0.0878	5.00	4,000	0.0653	0.0340	12.00	4,000
	1000	0.1223	0.0896	35.00	40,000	0.0694	0.0296	84.00	40,000
	2000	0.1223	0.0895	70.00	80,000	0.0693	0.0274	170.00	80,000
	4000	0.1221	0.0893	140.00	160,000	0.0693	0.0264	323.00	160,000
-		F3: $m = 100, n = 2000$				F4: $m = 250, n = 5000$			
ALG.	N	OBJ	DEV	TIME	ORC	OBJ	DEV	TIME	ORC
N-SA	100	0.1296	0.0735	1.00	200	0.1278	0.0800	3.00	200
	1000	0.1305	0.0709	6.00	1,100	0.1329	0.0808	15.00	1,100
	2000	0.1318	0.0812	11.00	2,100	0.1338	0.0834	29.00	2,100
	4000	0.1331	0.0834	21.00	4,100	0.1334	0.0831	56.00	4,100
E-SA	100	0.1277	0.0588	2.00	200	0.1153	0.0603	7.00	200
	1000	0.1281	0.0565	16.00	1,100	0.1312	0.0659	39.00	1,100
	2000	0.1287	0.0589	28.00	2,100	0.1312	0.0656	72.00	2,100
	4000	0.1303	0.0627	53.00	4,100	0.1310	0.0683	127.00	4,100
SAA	100	0.1310	0.0773	20.00	4,000	0.1253	0.0625	60.00	4,000
	1000	0.1294	0.0588	157.00	40,000	0.1291	0.0667	466.00	40,000
	2000	0.1304	0.0621	311.00	80,000	0.1284	0.0642	986.00	80,000
	4000	0.1301	0.0636	636.00	160,000	0.1293	0.0659	1885.00	160,000

4.3 A network planning problem with random demand

In the last experiment, we consider the so-called SSN problem of Sen, Doverspike, and Cosares [15]. This problem arises in telecommunications network design where the owner of the network sells private-line services between pairs of nodes in the network, and the demands are treated as random variables based on the historical demand patterns. The optimization problem is to decide where to add capacity to the network to minimize the expected rate of unsatisfied demands. Since this problem has been studied by several authors (see, e.g., [7, 15]), it could be interesting to compare the results. Another purpose of this experiment is to investigate the behavior of the SA method when one variance reduction technique, namely, the Latin Hyperplane Sampling (LHS), is applied.

The problem has been formulated as a two-stage stochastic linear programming as follows:

$$\begin{array}{ll} \min\limits_{x} & \{f(x) := \mathbb{E}[Q(x,\xi)]\} \\ \text{s.t.} & \sum_j x_j = b, x_j \geq 0, \end{array} \tag{4.3}$$

where x is the vector of capacities to be added to the arcs of the network, b (the budget) is the total amount of capacity to be added, ξ denotes the random demand, and $Q(x, \xi)$ represents the

Table 5: The variability for the stochastic max-flow problem

-		N-SA			E-SA			SAA		
		OBJ		AVG.	OBJ		AVG.	OBJ		AVG.
INST	N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
F2	1,000	0.0691	0.0004	3.11	0.0688	0.0006	4.62	0.0694	0.0003	90.15
F2	2,000	0.0694	0.0003	6.07	0.0692	0.0002	6.91	0.0695	0.0003	170.45

number of unserved requests. We have

$$\begin{array}{rll} Q(x,\tilde{\xi}) = & \min\limits_{s,f} & \sum_i s_i \\ & \text{s.t.} & \sum_i \sum_{r\in R(i)} A_{ir} f_{ir} \leq x + c, \\ & & \sum_{r\in R(i)} f_{ir} + s_i = \tilde{\xi}_i, \quad \forall i, \\ & & f_{ir} \geq 0, s_i \geq 0, \quad \forall i, r \in R(i). \end{array} \tag{4.4}$$

Here, $R(i)$ denotes a set of routes that can be used for connections associated with the node-pair i (Note that a static network-flow model is used in the formulation to simplify the problem); $\tilde{\xi}$ is a realization of the random variable ξ; the vectors A_{ir} are incidence vectors whose jth component a_{irj} is 1 if the link j belongs to the route r and is 0 otherwise; c is the vector of current capacities; f_{ir} is the number of connections associated with pair i using route $r \in R(i)$; s is the vector of unsatisfied demands for each request.

In the data set for SSN, there are total of 89 arcs and 86 point-to-point pairs; that is, the dimension of x is 89 and of ξ is 86. Each component of ξ is an independent random variable with a known discrete distribution. Specifically, there are between three and seven possible values for each component of ξ, giving a total of approximately 10^{70} possible complete demand scenarios.

The three methods, namely: N-SA, E-SA and SAA, and the same stepsize policy as discussed in Subsection 4.1, were applied for solving the SSN problem. Moreover, we compare these methods with or without using the Latin Hyperplane Sampling (LHS) technique. In the first test, each algorithm was run once for each test instance and the computational results are reported in Table 6, where N denotes the number of samples, 'OBJ' and 'DEV' represent the mean and standard deviation, respectively, of objective values of problem (4.3) as evaluated over $K = 10^4$ sample size at the approximated solutions yielded by the algorithms, 'TIME' is CPU seconds for obtaining the approximated solution, and 'ORC' stands for the number of calls to the stochastic oracle. Similar to the stochastic utility problem, we investigate the variability of these three methods by running each method for 100 times and computing the statistical results as shown in Table 7. Note that these tests for the SSN problem were conducted on a more powerful computer: Intel Xeon 1.86GHz with Red Hat Enterprize Linux.

This experiment shows that for a given sample size N, the solution quality for N-SA is close to that for SAA, meanwhile, the solution time of N-SA is much smaller than the latter one. However, for this particular instance, the improvement on the solution quality by using the Latin Hyperplane sampling is not significant, especially when a larger sample-size is applied. This result seems to be consistent with the observation in [7].

Table 6: SA vs. SAA on the SSN problem

-		Without LHS				With LHS			
ALG.	N	OBJ	DEV	TIME	ORC	OBJ	DEV	TIME	ORC
N-SA	100	11.0984	19.2898	1.00	200	10.1024	18.7742	1.00	200
	1,000	10.0821	18.3557	6.00	1100	10.0313	18.0926	7.00	1100
	2,000	9.9812	18.0206	12.00	2100	9.9936	17.9069	12.00	2100
	4,000	9.9151	17.9446	23.00	4100	9.9428	17.9934	22.00	4100
E-SA	100	10.9027	19.1640	1.00	200	10.3860	19.1116	1.00	200
	1,000	10.1268	18.6424	6.00	1100	10.0984	18.3513	6.00	1100
	2,000	10.0304	18.5600	12.00	2100	10.0552	18.4294	12.00	2100
	4,000	9.9662	18.6180	23.00	4100	9.9862	18.4541	23.00	4100
SAA	100	11.8915	19.4606	24.00	4,000	11.0561	20.4907	23.00	4000
	1,000	10.0939	19.3332	215.00	40,000	10.0488	19.4696	216.00	40,000
	2,000	9.9769	19.0010	431.00	80,000	9.9872	18.9073	426.00	80,000
	4,000	9.8773	18.9184	849.00	160,000	9.9051	18.3441	853.00	160,000

Table 7: The variability for the SSN problem

-		N-SA			E-SA			SAA		
		OBJ		AVG.	OBJ		AVG.	OBJ		AVG.
N	LHS	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
1,000	no	10.0624	0.1867	6.03	10.1730	0.1826	6.12	10.1460	0.2825	215.06
1,000	yes	10.0573	0.1830	6.16	10.1237	0.1867	6.14	10.0135	0.2579	216.10
2,000	no	9.9965	0.2058	11.61	10.0853	0.1887	11.68	9.9943	0.2038	432.93
2,000	yes	9.9978	0.2579	11.71	10.0486	0.2066	11.74	9.9830	0.1872	436.94

4.4 N-SA vs. E-SA

The data in Tables 3, 4, 6 demonstrate that with the same sample size N, the N-SA somehow outperforms the E-SA in terms of both the quality of approximate solutions and the running time. The difference, at the first glance, seems slim, and one could think that adjusting the SA algorithm to the "geometry" of the problem in question (in our case, to minimization over a standard simplex) is of minor importance. We, however, do believe that such a conclusion would be wrong. In order to get a better insight, let us come back to the stochastic utility problem. This test problem has an important advantage – we can easily compute the value of the objective $f(x)$ at a given candidate solution x analytically[1]. Moreover, it is easy to minimize $f(x)$ over the simplex – on a closest inspection, this problem reduces to minimizing an easy-to-compute *univariate* convex function, so that we can approximate the true optimal value f_* to high accuracy by Bisection. Thus, in the case in question we can compare solutions x generated by various algorithms in terms of their "true inaccuracy" $f(x) - f_*$, and this is the rationale behind our "Gaussian setup". We can now exploit the just outlined advantage of the stochastic utility problem for comparing properly N-SA and E-SA. In Table 8, we present the true values of the objective $f(x_*)$ at the approximate solutions x_* generated by N-SA and E-SA as applied to the instances L1 and L4 of the stochastic utility problem (cf. Table 3) along with the inaccuracies $f(x_*) - f_*$ and the Monte Carlo estimates $\widehat{f}(x_*)$ of $f(x_*)$

[1] Indeed, $(\xi_1, ..., \xi_n) \sim N(0, I_n)$, so that the random variable $\xi_x = \sum_i (a_i + \xi_i) x_i$ is normal with easily computable mean and variance, and since ϕ is piecewise linear, the expectation $f(x) = \mathbb{E}[\phi(\xi_x)]$ can be immediately expressed via the error function.

Table 8: N-SA vs. E-SA

METHOD	PROBLEM	$\widehat{f}(x_*), f(x_*)$	$f(x_*) - f_*$	TIME
N-SA, $N = 2,000$	L2: $n = 1000$	-5.9232/-5.9326	0.0113	2.00
E-SA, $N = 2,000$	L2	-5.8796/-5.8864	0.0575	7.00
E-SA, $N = 10,000$	L2	-5.9059/-5.9058	0.0381	13.00
E-SA, $N = 20,000$	L2	-5.9151/-5.9158	0.0281	27.00
N-SA, $N = 2,000$	L4: $n = 5000$	-5.5855/-5.5867	0.0199	6.00
E-SA, $N = 2,000$	L4	-5.5467/-5.5469	0.0597	10.00
E-SA, $N = 10,000$	L4	-5.5810/-5.5812	0.0254	36.00
E-SA, $N = 20,000$	L4	-5.5901/-5.5902	0.0164	84.00

obtained via 50,000-element samples. We see that the difference in the inaccuracy $f(x_*) - f_*$ of the solutions produced by the algorithms is much more significant than it is suggested by the data in Table 3 (where the actual inaccuracy is "obscured" by the estimation error and summation with f_*). Specifically, at the common for both algorithms sample size $N = 2,000$, the inaccuracy yielded by N-SA is 3 – 5 times less than the one for E-SA, and in order to compensate for this difference, one should increase the sample size for E-SA (and hence the running time) by factor 5 – 10. It should be added that in light of theoretical complexity analysis carried out in Example 2.1, the outlined significant difference in performances of N-SA and E-SA is not surprising; the surprising fact is that E-SA works at all.

4.5 Bilinear matrix game

We consider here a bilinear matrix game

$$\min_{x \in X} \max_{y \in Y} y^T A x,$$

where both feasible sets are the standard simplices in $\mathbb{R}^n$, i.e., $Y = X = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x \geq 0\}$. We consider two versions of the randomized Mirror SA algorithm (3.6), (3.7) for the saddle point problem: Euclidean Saddle Point SA (E-SA) which uses as ω_x and ω_y Euclidean distance generating function $\omega_x(x) = \frac{1}{2}\|x\|_2^2$. The other version of the method, which is referred to as the Non-Euclidean Saddle Point SA (N-SA), employs the entropy distance generating function defined in (2.49). To compare the two procedures we compute the corresponding approximate solutions tz_N after N iterations and compute the exact values of the error:

$$\epsilon(\tilde{z}_N) := \max_{y \in Y} y^T A \tilde{x}_N - \min_{x \in X} \tilde{y}_N^T A x, \quad i = 1, 2.$$

In our experiments we consider symmetric matrices A of two kinds. The matrices of the first family, parameterized by $\alpha > 0$, have the elements which obey the formula

$$A_{ij} := \left(\frac{i + j - 1}{2n - 1}\right)^\alpha, \;\; 1 \leq i, j \leq n.$$

The second family of matrices, which is also parameterized by $\alpha > 0$, contains the matrices with generic element

$$A_{ij} := \left(\frac{|i - j| + 1}{2n - 1}\right)^\alpha, \;\; 1 \leq i, j \leq n.$$

Table 9: SA for bilinear matrix games

	$E_2(2)$, $\epsilon(\tilde{z}_1) = 0.500$			$E_2(1)$, $\epsilon(\tilde{z}_1) = 0.500$			$E_2(0.5)$, $\epsilon(\tilde{z}_1) = 0.390$		
N-SA	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.
N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
100	0.0121	3.9 e-4	0.58	0.0127	1.9 e-4	0.69	0.0122	4.3 e-4	0.81
1,000	0.00228	3.7 e-5	5.8	0.00257	2.2 e-5	7.3	0.00271	4.5 e-5	8.5
2,000	0.00145	2.1 e-5	11.6	0.00166	1.0 e-5	13.8	0.00179	2.7 e-5	16.4
E-SA	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.
N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
100	0.00952	1.0 e-4	1.27	0.0102	5.1 e-5	1.77	0.00891	1.1 e-4	1.94
1,000	0.00274	1.3 e-5	11.3	0.00328	7.8 e-6	17.6	0.00309	1.6 e-5	20.9
2,000	0.00210	7.4 e-6	39.7	0.00256	4.6 e-6	36.7	0.00245	7.8 e-6	39.2

	$E_1(2)$, $\epsilon(\tilde{z}_1) = 0.0625$			$E_1(1)$, $\epsilon(\tilde{z}_1) = 0.125$			$E_1(0.5)$, $\epsilon(\tilde{z}_1) = 0.138$		
N-SA	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.
N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
100	0.00817	0.0016	0.58	0.0368	0.0068	0.66	0.0529	0.0091	0.78
1,000	0.00130	2.7 e-4	6.2	0.0115	0.0024	6.5	0.0191	0.0033	7.6
2,000	0.00076	1.6 e-4	11.4	0.00840	0.0014	11.7	0.0136	0.0018	13.8
E-SA	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.	$\epsilon(\tilde{z}_N)$		AVG.
N	MEAN	DEV	TIME	MEAN	DEV	TIME	MEAN	DEV	TIME
100	0.00768	0.0012	1.75	0.0377	0.0062	2.05	0.0546	0.0064	2.74
1,000	0.00127	2.2 e-4	17.2	0.0125	0.0022	19.9	0.0207	0.0020	18.4
2,000	0.00079	1.6 e-4	35.0	0.00885	0.0015	36.3	0.0149	0.0020	36.7

We use the notations $E_1(\alpha)$ and $E_2(\alpha)$ to refer to the experiences with the matrices of the first and second kind with parameter α. We present in Table 9 the results of experiments conducted for the matrices A of size $10^4 \times 10^4$. We have done 100 simulation runs in each experiment, we present the average error (column MEAN), standard deviation (column DAV) and the average running time (time which is necessary to compute the error of the solution is not taken into account). For comparison we also present the error of the initial solution $\tilde{z}_1 = (x_1, y_1)$.

Our basic observation is as follows: both Non-Euclidean SA (N-SA) and Euclidean SA (E-SA) algorithms succeed to reduce the error of solution reasonably fast. The mirror implementation is preferable as it is more efficient in terms of running time. For comparison, it takes MATLAB from 10 (for the simplest problem) to 35 seconds (for the hardest one) to compute just one answer $\mathsf{g}(x, y) = \begin{bmatrix} A^T y \\ -Ax \end{bmatrix}$ of the deterministic oracle.

References

[1] Azuma, K. Weighted sums of certain dependent random variables. *Tökuku Math. J.*, **19**, 357-367 (1967).

[2] Ben-Tal, A. and Nemirovski, A., Non-euclidean restricted memory level method for large-scale convex optimization, *Mathematical Programming*, **102**, 407-456 (2005).

[3] Benveniste, A., Métivier, M., Priouret, P., *Algorithmes adaptatifs et approximations stochastiques* , Masson, (1987). English trans. *Adaptive Algorithms and Stochastic Approximations*, Springer Verlag (1993).

[4] Chung, K.L., On a stochastic approximation method, *Ann. Math. Stat.* **25**, 463-483 (1954).

[5] Goldberg, A.V. and Tarjan, R.E., A New Approach to the Maximum Flow Problem, *Journal of ACM*, **35**, 921-940 (1988).

[6] Kleywegt, A. J., Shapiro, A. and Homem-de-Mello, T., The sample average approximation method for stochastic discrete optimization, *SIAM J. Optimization*, **12**, 479-502 (2001).

[7] Linderoth, J., Shapiro, A. and Wright, S., The empirical behavior of sampling methods for stochastic programming, *Annals of Operations Research*, **142**, 215-241 (2006).

[8] 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**, 47–56 (1999).

[9] Nemirovskii, A., and Yudin, D. "On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions." (in Russian) - *Doklady Akademii Nauk SSSR*, v. 239 (1978) No. 5 (English translation: *Soviet Math. Dokl.* v. 19 (1978) No. 2)

[10] Nemirovski, A., Yudin, D., *Problem complexity and method efficiency in optimization*, Wiley-Interscience Series in Discrete Mathematics, John Wiley, XV, 1983.

[11] Polyak, B.T., New stochastic approximation type procedures, *Automat. i Telemekh.*, 7 (1990), 98-107.

[12] Polyak, B.T. and Juditsky, A.B., Acceleration of stochastic approximation by averaging, *SIAM J. Control and Optimization*, 30 (1992), 838-855.

[13] Robbins, H. and Monro, S., A stochastic spproximation method, *Annals of Math. Stat.*, 22 (1951), 400-407.

[14] Sacks, J., Asymptotic distribution of stochastic approximation, *Ann. Math. Stat.*, **29**, 373-409 (1958).

[15] Sen, S., Doverspike, R.D. and Cosares, S., Network Planning with Random Demand, *Telecommunication Systems*, **3**, 11-30 (1994).

[16] Shapiro, A., Monte Carlo sampling methods, in: Ruszczyński, A. and Shapiro, A., (Eds.), *Stochastic Programming*, Handbook in OR & MS, Vol. 10, North-Holland Publishing Company, Amsterdam, 2003.

[17] 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.

[18] Strassen, V., The existence of probability measures with given marginals, *Annals of Mathematical Statistics*, **38**, 423–439 (1965).

[19] Verweij, B., Ahmed, S., Kleywegt, A.J., Nemhauser, G. and Shapiro, A., The sample average approximation method applied to stochastic routing problems: a computational study, *Computational Optimization and Applications*, **24**, 289–333 (2003).

5 Appendix

Proof of Lemma 2.1. Let $x \in X^o$ and $v = P_x(y)$; note that v is of the form $\mathrm{argmin}_{z\in X}[\omega(z) + p^T z]$ and thus $v \in X^o$, so that ω is differentiable at v. As $\nabla_v V(x,v) = \nabla\omega(v) - \nabla\omega(x)$, the optimality conditions for (2.26) imply that

$$(\nabla\omega(v) - \nabla\omega(x) + y)^T(v-u) \quad \leq \quad 0 \quad \forall u \in X. \tag{5.1}$$

For $u \in X$ we therefore have

$$\begin{array}{rcl} V(v,u) - V(x,u) & = & [\omega(u) - \nabla\omega(v)^T(u-v) - \omega(v)] - [\omega(u) - \nabla\omega(x)^T(u-x) - \omega(x)] \\ & = & (\nabla\omega(v) - \nabla\omega(x) + y)^T(v-u) + y^T(u-v) \\ & & -[\omega(v) - \nabla\omega(x)^T(v-x) - \omega(x)] \\ \text{[due to (5.1)]} & \leq & y^T(u-v) - V(x,v). \end{array}$$

By Young's inequality[2] we have

$$y^T(x-v) \leq \frac{\|y\|_*^2}{2\alpha} + \frac{\alpha}{2}\|x-v\|^2,$$

while $V(x,v) \geq \frac{\alpha}{2}\|x-v\|^2$, due to the strong convexity of $V(x,\cdot)$. We get

$$V(v,u) - V(x,u) \leq y^T(u-v) - V(x,v) = y^T(u-x) + y^T(x-v) - V(x,v) \leq y^T(u-x) + \frac{\|y\|_*^2}{2\alpha},$$

as required in (2.28).

Entropy as a distance-generating function on the standard simplex. The only property which is not immediately evident is that the entropy $w(x) := \sum_{i=1}^n x_i \ln x_i$ is strongly convex, modulus 1 with respect to $\|\cdot\|_1$-norm, on the standard simplex $X := \left\{x \in \mathbb{R}^n : x \geq 0,\ \sum_{i=1}^n x_i\right\}$. We are in the situation where $X^o = \{x \in X : x > 0\}$, and in order to establish the property in question it suffices to verify that $h^T\nabla^2\omega(x)h \geq \|h\|_1^2$ for every $x \in X^o$. Here is the computation:

$$\left[\sum_i |h_i|\right]^2 = \left[\sum_i (x_i^{-1/2}|h_i|)x_i^{1/2}\right]^2 \leq \left[\sum_i h_i^2 x_i^{-1}\right]\left[\sum_i x_i\right] = \sum_i h_i^2 x_i^{-1} = h^T\nabla^2\omega(x)h,$$

where the inequality follows by Cauchy inequality.

[2] For any $u, v \in \mathbb{R}^n$ we have by the definition of the dual norm that $\|u\|_*\|v\| \geq u^Tv$ and hence $(\|u\|_*^2/\alpha + \alpha\|v\|^2)/2 \geq \|u\|_*\|v\| \geq u^Tv$.

Proof of Lemma 3.1. By (2.28) we have for any $u \in Z$ that

$$\gamma_t(z_t - u)^T \mathsf{G}(z_t, \xi_t) \le V(z_t, u) - V(z_{t+1}, u) + \frac{\gamma_t^2}{2}\|\mathsf{G}(z_t, \xi_t)\|_*^2 \tag{5.2}$$

(recall that we are in the situation of $\alpha = 1$). This relation implies that for every $u \in Z$ one has

$$\gamma_t(z_t - u)^T \mathsf{g}(z_t) \quad \le \quad V(z_t, u) - V(z_{t+1}, u) + \frac{\gamma_t^2}{2}\|\mathsf{G}(z_t, \xi_t)\|_*^2 - \gamma_t(z_t - u)^T \Delta_t, \tag{5.3}$$

where $\Delta_t := \mathsf{G}(z_t, \xi_t) - \mathsf{g}(z_t)$. Summing up these inequalities over $t = 1, ..., j$, we get

$$\sum_{t=1}^{j} \gamma_t(z_t - u)^T \mathsf{g}(z_t) \quad \le \quad V(z_1, u) - V(z_{t+1}, u) + \sum_{t=1}^{j} \frac{\gamma_t^2}{2}\|\mathsf{G}(z_t, \xi_t)\|_*^2 - \sum_{t=1}^{j} \gamma_t(z_t - u)^T \Delta_t.$$

Now we need the following simple lemma.

Lemma 5.1 *Let $\zeta_1, ..., \zeta_j$ be a sequence of elements of $\mathbb{R}^{n+m}$. Define the sequence v_t, $t = 1, 2, ...$ in Z^o as follows: $v_1 \in Z^o$ and*

$$v_{t+1} = P_{v_t}(\zeta_t),\ 1 \le t \le j.$$

Then for any $u \in Z$ the following holds

$$\sum_{t=1}^{j} \zeta_t^T(v_t - u) \le V(v_1, u) + \tfrac{1}{2}\sum_{t=1}^{j} \|\zeta_t\|_*^2. \tag{5.4}$$

Proof. Using the bound (2.28) of Lemma 2.1 with $y = \zeta_t$ and $x = v_t$ (so that $v_{t+1} = P_{v_t}(\zeta_t)$) and recalling that we are in the situation of $\alpha = 1$, we obtain for any $u \in Z$:

$$V(v_{t+1}, u) \le V(v_t, u) + \zeta_t^T(u - v_t) + \frac{\|\zeta_t\|_*^2}{2}.$$

Summing up from $t = 1$ to $t = j$ we conclude that

$$V(v_{j+1}, u) \le V(v_1, u) + \sum_{t=1}^{j} \zeta_t^T(u - v_t) + \sum_{t=1}^{j} \frac{\|\zeta_t\|_*^2}{2},$$

which implies (5.4) due to $V(v, u) \ge 0$ for any $v \in Z^o, u \in Z$. ▮

Applying Lemma 5.1 with $v_1 = z_1$, $\zeta_t = -\gamma_t \Delta_t$:

$$\forall u \in Z: \quad \sum_{t=1}^{j} \gamma_t \Delta_t^T(u - v_t) \le V(z_1, u) + \frac{1}{2}\sum_{t=1}^{j} \gamma_t^2 \|\Delta_t\|_*^2. \tag{5.5}$$

Observe that

$$\mathbb{E}\|\Delta_t\|_*^2 \le 4\mathbb{E}\|\mathsf{G}(z_t, \xi_t)\|_*^2 \le 4\left(\frac{2D_{\omega_x,X}^2}{\alpha_x} M_{*,x}^2 + \frac{2D_{\omega_y,Y}^2}{\alpha_y} M_{*,y}^2\right) = 4M_*^2,$$

so that when taking the expectation of both sides of (5.5) we get

$$\mathbb{E}\sup_{u\in Z}\left(\sum_{t=1}^{j}\gamma_t\Delta_t^T(u-v_t)\right)\leq 1+2M_*^2\sum_{t=1}^{j}\gamma_t^2 \tag{5.6}$$

(recall that $V(z_1,\cdot)$ is bounded by 1 on Z). Now we proceed exactly as in Section 2.2: we sum up (5.3) from $t=1$ to j to obtain

$$\begin{aligned}\sum_{t=1}^{j}\gamma_t(z_t-u)^T\mathsf{g}(z_t) &\leq V(z_1,u)+\sum_{t=1}^{j}\frac{\gamma_t^2}{2}\|\mathsf{G}(z_t,\xi_t)\|_*^2-\sum_{t=1}^{j}\gamma_t(z_t-u)^T\Delta_t\\ &= V(z_1,u)+\sum_{t=1}^{j}\frac{\gamma_t^2}{2}\|\mathsf{G}(z_t,\xi_t)\|_*^2-\sum_{t=1}^{j}\gamma_t(z_t-v_t)^T\Delta_t+\sum_{t=1}^{j}\gamma_t(u-v_t)^T\Delta_t. \end{aligned}\tag{5.7}$$

When taking into account that z_t and v_t are deterministic functions of $\xi_{[t-1]}=(\xi_1,...,\xi_{t-1})$ and that the conditional expectation of Δ_t, $\xi_{[t-1]}$ being given, vanishes, we conclude that $\mathbb{E}[(z_t-v_t)^T\Delta_t]=0$. We take now suprema in $u\in Z$ and then expectations on both sides of (5.7):

$$\begin{aligned}\mathbb{E}\left[\sup_{u\in Z}\sum_{t=1}^{j}\gamma_t(z_t-u)^T\mathsf{g}(z_t)\right] &\leq \sup_{u\in Z}V(z_1,u)+\sum_{t=1}^{j}\frac{\gamma_t^2}{2}\mathbb{E}\|\mathsf{G}(z_t,\xi_t)\|_*^2+\sup_{u\in Z}\sum_{t=1}^{j}\gamma_t(u-v_t)^T\Delta_t\\ \text{[by (5.6)]}\quad &\leq 1+\frac{M_*^2}{2}\sum_{t=1}^{j}\gamma_t^2+\left[1+2M_*^2\sum_{t=1}^{j}\gamma_t^2\right]=2+\frac{5}{2}M_*^2\sum_{t=1}^{j}\gamma_t^2.\end{aligned}$$

and we arrive at (3.9). ▮

Proof of Propositions 2.1 and 3.1. We provide here the proof of Proposition 3.1 only. The proof of Proposition 2.1 follows the same lines and can be easily reconstructed using the bound (2.34) instead of the relations (5.5) and (5.7) in the proof below.

First of all, with M_* given by (3.5) one has

$$\forall(z\in Z):\mathbb{E}\left[\exp\{\|\mathsf{G}(z,\xi)\|_*^2/M_*^2\}\right]\leq\exp\{1\}. \tag{5.8}$$

Indeed, setting $p_x=\frac{2D^2_{\omega_x,X}M^2_{*,x}}{\alpha_x M_*^2}$, $p_y=\frac{2D^2_{\omega_y,Y}M^2_{*,y}}{\alpha_y M_*^2}$ we have $p_x+p_y=1$, whence, invoking (3.4),

$$\mathbb{E}\left[\exp\{\|\mathsf{G}(z,\xi)\|_*^2/M_*^2\}\right]=\mathbb{E}\left[\exp\{p_x\|\mathsf{G}_x(z,\xi)\|_{*,x}^2/M_{*,x}^2+p_y\|\mathsf{G}_y(z,\xi)\|_{*,y}^2/M_{*,y}^2\}\right],$$

and (5.8) follows from (3.13) by the Hölder inequality.

Setting $\Gamma_N=\sum_{t=1}^{N}\gamma_t$ and using the notation from the proof of Lemma 3.1, relations (3.8), (5.5), (5.7) combined with the fact that $V(z_1,u)\leq 1$ for $u\in Z$, imply that

$$\Gamma_N\epsilon_\phi(\tilde{z}_N)\leq 2+\underbrace{\sum_{t=1}^{N}\frac{\gamma_t^2}{2}\left[\|\mathsf{G}(z_t,\xi_t)\|_*^2+\|\Delta_t\|_*^2\right]}_{\alpha_N}+\underbrace{\sum_{t=1}^{N}\gamma_t(v_t-z_t)^T\Delta_t}_{\beta_N}. \tag{5.9}$$

Now, from (5.8) it follows straightforwardly that

$$\mathbb{E}[\exp\{\|\Delta_t\|_*^2/(2M_*)^2\}]\leq\exp\{1\},\quad \mathbb{E}[\exp\{\|\mathsf{G}(z_t,\xi_t)\|_*^2/M_*^2\}]\leq\exp\{1\}, \tag{5.10}$$

which in turn implies that

$$\mathbb{E}[\exp\{\alpha_N/\sigma_\alpha\}] \le \exp\{1\}, \quad \sigma_\alpha = \frac{5}{2}M_*^2\sum_{t=1}^N \gamma_t^2, \tag{5.11}$$

and therefore, by Markov inequality,

$$\forall(\Omega > 0): \ \ \mathrm{Prob}\{\alpha_N \ge (1+\Omega)\sigma_\alpha\} \le \exp\{-\Omega\}. \tag{5.12}$$

Indeed, we have by (5.8)

$$\|\mathsf{g}(z_t)\|_* = \|\mathbb{E}[\mathsf{G}(z_t,\xi_t)|\xi_{[t-1]}]\|_* \le \sqrt{\mathbb{E}(\|\mathsf{G}(z_t,\xi_t)\|_*^2|\xi_{[t-1]})} \le M_*,$$

and

$$\|\Delta_t\|_*^2 = \|\mathsf{G}(z_t,\xi_t) - \mathsf{g}(z_t)\|_*^2 \le (\|\mathsf{G}(z_t,\xi_t)\|_* + \|\mathsf{g}(z_t)\|_*)^2 \le 2\|\mathsf{G}(z_t,\xi_t)\|_*^2 + 2M_*^2,$$

what implies that

$$\alpha_N \le \sum_{t=1}^N \frac{\gamma_t^2}{2}\left[3\|\mathsf{G}(z_t,\xi_t)\|_*^2 + 2M_*^2\right].$$

Further, by the Hölder inequality we have from (5.8):

$$\mathbb{E}\left[\exp\left\{\frac{\gamma_t^2\left[\frac{3}{2}\|\mathsf{G}(z_t,\xi_t)\|_*^2 + M_*^2\right]}{\frac{5}{2}\gamma_t^2M_*^2}\right\}\right] \le \exp(1).$$

Observe that if $r_1, ..., r_i$ are nonnegative random variables such that $\mathbb{E}[\exp\{r_t/\sigma_t\}] \le \exp\{1\}$ for some deterministic $\sigma_t > 0$, then, by convexity of the exponent, $w(s) = \exp\{s\}$,

$$\mathbb{E}\left[\exp\left\{\frac{\sum_{t\le i} r_t}{\sum_{t\le i}\sigma_t}\right\}\right] \le \mathbb{E}\left[\sum_{t\le i}\frac{\sigma_t}{\sum_{\tau\le i}\sigma_\tau}\exp\{r_t/\sigma_t\}\right] \le \exp\{1\}. \tag{5.13}$$

Now applying (5.13) with $r_t = \gamma_t^2\left[\frac{3}{2}\|\mathsf{G}(z_t,\xi_t)\|_*^2 + M_*^2\right]$ and $\sigma_t = \frac{5}{2}\gamma_t^2M_*^2$ we obtain (5.11).

Now let $\zeta_t = \gamma_t(v_t - z_t)^T\Delta_t$. Observing that v_t, z_t are deterministic functions of $\xi_{[t-1]}$, while $\mathbb{E}[\Delta_t|\xi_{[t-1]}] = 0$, we see that the sequence $\{\zeta_t\}_{t=1}^N$ of random real variables forms a martingale-difference. Besides this, by strong convexity of ω with modulus 1 w.r.t. $\|\cdot\|$ and due to $D_{\omega,Z} \le 1$ we have

$$u \in Z \Rightarrow 1 \ge V(z_1,u) \ge \frac{1}{2}\|u - z_1\|^2,$$

whence the $\|\cdot\|$-diameter of Z does not exceed $2\sqrt{2}$, so that $|\zeta_t| \le 2\sqrt{2}\gamma_t\|\Delta_t\|_*$, and therefore

$$\mathbb{E}[\exp\{|\zeta_t|^2/(32\gamma_t^2M_*^2)\}|\xi_{[t-1]}] \le \exp\{1\}$$

by (5.10). Applying Cramer's deviation bound, we arrive at

$$\forall\Omega > 0: \ \ \mathrm{Prob}\left\{\beta_N > 4\Omega M_*\sqrt{\textstyle\sum_{t=1}^N\gamma_t^2}\right\} \le \exp\{-\Omega^2/4\}. \tag{5.14}$$

Indeed, for $0 \le \gamma$, setting $\sigma_t = 4\sqrt{2}\gamma_tM_*$ and taking into account that ζ_t is a deterministic function of $\xi_{[t]}$ with $\mathbb{E}[\zeta_t|\xi_{[t-1]}] = 0$ and $\mathbb{E}[\exp\{\zeta_t^2/\sigma_t^2\}|\xi_{[t-1]}] \le \exp\{1\}$, we have

$$\begin{array}{l}
0 < \gamma\sigma_t \le 1 \Rightarrow \ \ (\text{as } e^x \le x + e^{x^2}) \\
\mathbb{E}[\exp\{\gamma\zeta_t\}|\xi_{[t-1]}] \le \mathbb{E}[\exp\{\gamma^2\zeta_t^2\}|\xi_{[t-1]}] \le \mathbb{E}[(\exp\{\zeta_t^2/\sigma_t^2\})^{\gamma^2\sigma_t^2}|\xi_{[t-1]}] \le \exp\{\gamma^2\sigma_t^2\}; \\
\gamma\sigma_t > 1 \Rightarrow \\
\mathbb{E}[\exp\{\gamma\zeta_t\}|\xi_{[t-1]}] \le \mathbb{E}[\exp\{[\frac{1}{2}\gamma^2\sigma_t^2 + \frac{1}{2}\zeta_t^2/\sigma_t^2\}|\xi_{[t-1]}] \le \exp\{\frac{1}{2}\gamma^2\sigma_t^2 + \frac{1}{2}\} \le \exp\{\gamma^2\sigma_t^2\}
\end{array}$$

that is, in both cases $\mathbb{E}[\exp\{\gamma\zeta_t\}|\xi_{[t-1]}] \leq \exp\{\gamma^2\sigma_t^2\}$. Therefore

$$\mathbb{E}[\exp\{\gamma\beta_i\}] = \mathbb{E}\left[\exp\{\gamma\beta_{i-1}\}\mathbb{E}[\exp\{\gamma\zeta_i\}|\xi_{[i-1]}]\right] \leq \exp\{\gamma^2\sigma_i^2\}\mathbb{E}[\exp\{\gamma\beta_{i-1}\}],$$

whence

$$\mathbb{E}[\exp\{\gamma\beta_N\}] \leq \exp\{\gamma^2\sum_{t=1}^N \sigma_t^2\},$$

and thus by Markov inequality for every $\Omega > 0$ it holds

$$\text{Prob}\left\{\beta_N > \Omega\sqrt{\sum_{t=1}^N \sigma_t^2}\right\} \leq \exp\left\{\gamma^2\sum_{t=1}^N \sigma_t^2\right\}\exp\left\{-\gamma\Omega\sqrt{\sum_{t=1}^N \sigma_t^2}\right\}.$$

When choosing $\gamma = \frac{1}{2}\Omega\left(\sum_{t=1}^N \sigma_t^2\right)^{-1/2}$ we arrive at (5.14).

Combining (5.9), (5.10) and (5.14), we get for any positive Ω and Θ:

$$\text{Prob}\left\{\Gamma_N\,\epsilon_\phi(\tilde{z}_t) > 2 + \frac{5}{2}(1+\Omega)M_*^2\sum_{t=1}^N\gamma_t^2 + 4\sqrt{2}\Theta M_*\sqrt{\sum_{t=1}^N\gamma_t^2}\right\} \leq \exp\{-\Omega\} + \exp\{-\Theta^2/4\}.$$

When setting $\Theta = 2\sqrt{\Omega}$ and substituting (3.10) we obtain (3.14). ▮

Proof of Proposition 3.2 As in the proof of Proposition 3.1, when setting $\Gamma_N = \sum_{t=1}^N \gamma_t$ and using the relations (3.8), (5.5), (5.7), combined with the fact that $\|\mathsf{G}(z,\xi_y)\|_* \leq M_*$, we obtain

$$\begin{aligned}\Gamma_N\,\epsilon_\phi(\tilde{z}_N) &\leq 2 + \sum_{t=1}^N \frac{\gamma_t^2}{2}\left[\|\mathsf{G}(z_t,\xi_t)\|_*^2 + \|\Delta_t\|_*^2\right] + \sum_{t=1}^N \gamma_t(v_t - z_t)^T\Delta_t \\ &\leq 2 + \frac{5}{2}M_*^2\sum_{t=1}^N\gamma_t^2 + \underbrace{\sum_{t=1}^N \gamma_t(v_t - z_t)^T\Delta_t}_{\alpha_N}. \qquad (5.15)\end{aligned}$$

Recall that by definition of Δ_t, $\|\Delta_t\|_* = \|\mathsf{G}(z_t,\xi_t) - \mathsf{g}(z_t)\|_* \leq \|\mathsf{G}(z_t,\xi_t)\| + \|\mathsf{g}(z_t)\|_* \leq 2M_*$.

Note that $\zeta_t = \gamma_t(v_t - z_t)^T\Delta_t$ is a bounded martingale-difference, i.e., $\mathbb{E}(\zeta_t|\xi_{[t-1]}) = 0$, and $|\zeta_t| \leq 4\gamma_t\overline{M}$ (here $\overline{M}$ is defined in (3.27)). Then by Azuma-Hoeffding's inequality [1] for any $\Omega \geq 0$:

$$\text{Prob}\left(\alpha_N > 4\Omega\overline{M}\sqrt{\sum_{t=1}^N\gamma_t^2}\right) \leq e^{-\Omega^2/2}. \qquad (5.16)$$

Indeed, let us denote $v_t = (v_t^{(x)}, v_t^{(y)})$ and $\Delta_t = (\Delta_t^{(x)}, \Delta_t^{(y)})$. When taking into account that $\|v_t^{(x)}\|_1 \leq 1$, $\|v_t^{(y)}\|_1 \leq 1$, and $\|x_t\|_1 \leq 1$, $\|y_t\|_1 \leq 1$, we conclude that

$$\begin{aligned}|(v_t - z_t)^T\Delta_t| &\leq |(v_t^{(x)} - x_t)^T\Delta_t^{(x)}| + |(v_t^{(y)} - y_t)^T\Delta_t^{(y)}| \\ &\leq 2\|\Delta_t^{(x)}\|_\infty + 2\|\Delta_t^{(y)}\|_\infty \leq 4\max_{1\leq j\leq m}\|A^j + b\|_\infty + 4\max_{1\leq j\leq n}\|A_j + c\|_\infty = 4\overline{M}.\end{aligned}$$

We conclude from (5.15) and (5.16) that

$$\text{Prob}\left(\Gamma_N\,\epsilon_\phi(\tilde{z}_N) > 2 + \frac{5}{2}M_*^2\sum_{t=1}^N\gamma_t^2 + 4\Omega\overline{M}\sqrt{\sum_{t=1}^N\gamma_t^2}\right) \leq e^{-\Omega^2/2},$$

and the bound (3.26) of the proposition can be easily obtained by substituting the constant stepsizes γ_t as defined in (3.10). ∎