Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

MIRROR DESCENT STOCHASTIC APPROXIMATION FOR COMPUTING WORST-CASE STOCHASTIC INPUT MODELS

Soumyadip Ghosh
Math Sciences Department
T.J. Watson IBM Research Center
Yorktown Heights, NY 10598, USA

Henry Lam
Department of Industrial and Operations Engineering
University of Michigan
1205 Beal Ave.
Ann Arbor, MI 48109, USA

ABSTRACT

Performance analysis via stochastic simulation is often subject to input model uncertainty, meaning that the input model is unknown and needs to be inferred from data. Motivated especially from situations with limited data, we consider a worst-case analysis to handle input uncertainty by representing the partially available input information as constraints and solving a worst-case optimization problem to obtain a conservative bound for the output. In the context of i.i.d. input processes, such approach involves simulation-based nonlinear optimizations with decision variables being probability distributions. We explore the use of a specialized class of mirror descent stochastic approximation (MDSA) known as the entropic descent algorithm, particularly effective for handling probability simplex constraints, to iteratively solve for the local optima. We show how the mathematical program associated with each iteration of the MDSA algorithm can be efficiently computed, and carry out numerical experiments to illustrate the performance of the algorithm.

1 INTRODUCTION

We consider stochastic simulation with uncertain input specifications. This arises in many contexts when only finite data is available for calibrating the input model. Conventional approaches to handle such uncertainty include goodness-of-fit tests (e.g. Banks et al. 2000), Bayesian model selection or averaging (e.g. Chick 2001), and bootstrapping (e.g. Barton et al. 2013). In this paper, we consider an alternative, worst-case approach, which can be particularly relevant when data is limited or unavailable, or when the modeler wants to assess risk beyond what data could tell. This approach comprises solving optimization programs on the space of input models, with constraints representing the partially known information on these inputs. The optimal values of these programs give the worst- or the best-case performance measures under the uncertainty.

More concretely, consider a stochastic simulation problem of computing $E_{\mathbf{p}}[h(\mathbf{X})]$, where $\mathbf{X} = (X_1, \ldots, X_T)$ are the i.i.d. replications from a common input model represented by the probability distribution $\mathbf{p}$. In other words, $X_t \overset{i.i.d.}{\sim} \mathbf{p}$. $h(\cdot)$ is some function that captures the system logic applied to $\mathbf{X}$, and $E_{\mathbf{p}}[\cdot]$ denotes the expectation with respect to the i.i.d. replications of X_t's each with distribution $\mathbf{p}$. As an example, one can think of $\mathbf{X}$ as a sequence of interarrival or service times, and $\mathbf{p}$ is the corresponding interarrival or service time distribution. $h(\mathbf{X})$ can be the indicator function of long waiting time for customer T, so that $E_{\mathbf{p}}[h(\mathbf{X})]$ is the corresponding long waiting time probability.

Our premise is that $\mathbf{p}$ is unknown, but some information is available. These information can be represented by constraints on $\mathbf{p}$, denoted $\mathscr{A}$. Our goal is to perform a worst-case analysis to calculate

$$\min_{\mathbf{p}\in\mathscr{A}} E_{\mathbf{p}}[h(\mathbf{X})] \quad \text{or} \quad \max_{\mathbf{p}\in\mathscr{A}} E_{\mathbf{p}}[h(\mathbf{X})]. \tag{1}$$

For instance, $\mathscr{A}$ can specify that the first two moments of $\mathbf{p}$ are within certain ranges. In this case, we can write

$$\mathscr{A} = \{\mathbf{p} : \underline{\mu} \leq E_{\mathbf{p}} X_1 \leq \overline{\mu},\ \underline{\sigma} \leq E_{\mathbf{p}} X_1^2 \leq \overline{\sigma}\}$$

for some numbers $\underline{\mu} \leq \overline{\mu}$ and $\underline{\sigma} \leq \overline{\sigma}$.

As another example, suppose that the modeler adopts a baseline model $\mathbf{b}$, e.g. exponential distribution, and believes that it is close to the truth. Then we can use the constraint

$$\mathscr{A} = \{\mathbf{p} : d(\mathbf{p}, \mathbf{b}) \leq \eta\}$$

where $d(\cdot,\cdot)$ is some statistical distance defined between two distributions, i.e. a notion of distance that is not restricted to any specific parametric models, and $\eta > 0$ is a threshold parameter that describes the level of uncertainty.

The key is that in both examples, the modeler represents the information on the input model nonparametrically, without committing to a specific parametric class that can be potentially misspecified. Later in this paper, we shall use the above two examples of $\mathscr{A}$ to illustrate our optimization procedure.

The worst-case formulation like (1) is inspired from the literature of distributionally robust optimization (DRO). This literature considers decision-making under ambiguity of the underlying probabilistic models (e.g. Delage and Ye 2010, Goh and Sim 2010, Ben-Tal et al. 2013), often cast in the form of minimax problems. Glasserman and Xu (2014) further studies the solutions to these problems via simulation.

However, unlike the standard DRO literature that possess convex formulations and linear (or stage-wise linear) objectives, the objective function in (1) is not linear, or even convex in $\mathbf{p}$ in general, since computing the expectation $E_{\mathbf{p}}[\cdot]$ actually involves a T-fold convolution of $\mathbf{p}$ due to the i.i.d. structure. The objective function in (1) is also accessible only via simulation in our context of interest. These challenges obstruct the direct usage of previous techniques. Moreover, because of non-convexity, we would set our goal as to obtain a local optimum for (1).

Formulation (1) also faces two additional challenges. The first is that $\mathbf{p}$ can be high-dimensional. In fact, for a continuous input distribution, $\mathbf{p}$ is infinite-dimensional. In this paper we shall restrict ourselves to discrete finite-support distributions, with the understanding that discretization is used a priori to handle continuous distributions. Second, (1) consists of constraints that can be computationally costly to various extents.

Ghosh and Lam (2015) discusses some of the above challenges and proposes a constrained stochastic approximation (SA) algorithm based on the Frank-Wolfe (FW) method in nonlinear programming (which they call FWSA), by successively solving linear-objective subprograms that give the best feasible direction at each iteration. They show almost sure convergence and analyze the convergence rate of the algorithm.

This paper describes an alternative to FWSA by using the idea of mirror descent (MD) (e.g. Nemirovski and Yudin 1983, Nemirovski et al. 2009). In fact, we will focus on a specialized class of mirror descent stochastic approximation (MDSA) that is based on a so-called entropic distance-generating function. The deterministic version of this algorithm is known as the entropic descent algorithm (EDA) (e.g. Beck and Teboulle 2003). The motivation of this alternative is two-fold. The first is that MD method, and in particular EDA, is known to scale favorably with the decision space's dimension when the space is a probability simplex. Since we consider potentially discrete distributions that have many support points (or very fine discretization of continuous distributions), this property of MD is attractive for our problem. While theoretical guarantees of this sort have been shown only for convex programs, our preliminary experimental results show the algorithm works well for some potentially non-convex settings. Second, unlike FWSA, MDSA does not require a growing sample size for gradient estimation. This can potentially save huge computational effort as iteration increases.

The rest of the paper is organized as follows. Section 2 introduces MDSA. Section 3 discusses gradient estimation. Section 4 analyzes the solution to the mathematical program required in each iteration of MDSA. Finally, Section 5 shows some numerical results for an $M/GI/1$ example.

2 MIRROR DESCENT STOCHASTIC APPROXIMATION

The MD procedure was first proposed by Nemirovski and Yudin (1983) for convex deterministic optimization. One of its earlier motivations is to define a proper gradient descent type algorithm for decision variable that lies on function spaces. In general, the gradient of the objective lies in the dual of the decision space, which is different from the primal, and so the usual iteration routine by moving the current solution along the gradient direction is not well-defined (the standard Euclidean space is an exception because it is isometric to its dual space). MD resolves this issue by first mapping the solution into the dual space, followed by applying the gradient descent in the dual space and at the end mapping back into the primal. This method can as well be used for decision variables in the Euclidean space: By appropriately defining the space and its norm that are pertained to the geometry of the constraints, it can achieve a rate of convergence that is favorably scaled with the dimension of the problem.

We first give a general description of MD, focusing on a generic minimization problem $\min_{x\in\mathscr{X}} Z(x)$. Adopting Nemirovski et al. (2009), we first define a distance-generating function $\omega(\cdot)$ and a norm $\|\cdot\|$ on the space $\mathscr{X}$. The function $\omega(\cdot)$ needs to be strongly convex with respect to the norm $\|\cdot\|$, meaning that

$$\omega(y)-\omega(x)\geq\nabla\omega(x)'(y-x)+\frac{\alpha}{2}\|y-x\|^2$$

for some constant $\alpha>0$ and any $x,y\in\mathscr{X}$, which is equivalent to

$$(y-x)'(\nabla\omega(y)-\nabla\omega(x))\geq\alpha\|y-x\|^2.$$

Next we define the prox-function $V(\cdot,\cdot)$ as

$$V(x,z)=\omega(z)-\omega(x)-\nabla\omega(x)'(z-x). \tag{2}$$

In other words, $V(\cdot,\cdot)$ is the remainder of the first order approximation of $\omega(\cdot)$. We then define the prox-mapping $\text{prox}_x(y)$ as

$$\text{prox}_x(y)=\text{argmin}_{z\in\mathscr{X}}\, y'(z-x)+V(x,z).$$

For an optimization $\min_{x\in\mathscr{X}} Z(x)$, the MD method iteratively updates x_{k+1} from x_k by using $x_{k+1}=\text{prox}_{x_k}(\gamma_k\nabla Z(x_k))$ where γ_k is some step size. If $\nabla Z(x_k)$ can only be assessed through unbiased samples, then an estimate $\widehat{\nabla Z}(x_k)$ is used, in which case it becomes MDSA.

We put the above discussion into our context. Our objective function is $Z(\mathbf{p})=E_{\mathbf{p}}[h(\mathbf{X})]$. We consider $\mathbf{p}$ in the probability simplex on n fixed support points $\{u_1,\ldots,u_n\}$, denoted by $\mathscr{P}=\{(p_1,\ldots,p_n):\sum_{i=1}^n p_i=1, p_i\geq 0 \text{ for all } i=1,\ldots,n\}$. The feasible region $\mathscr{A}\subset\mathscr{P}$ is our space in consideration. We shall use the entropic distance-generating function

$$\omega(\mathbf{p})=-\sum_{i=1}^n p_i\log p_i. \tag{3}$$

The corresponding prox-function is

$$V(\mathbf{p},\mathbf{q})=\sum_{i=1}^n q_i\log\frac{q_i}{p_i} \tag{4}$$

where $\mathbf{p}=(p_1,\ldots,p_n),\mathbf{q}=(q_1,\ldots,q_n)\in\mathscr{A}$, which can be seen by plugging (3) into (2). The MD iteration using the prox-mapping is

$$\mathbf{p}_{k+1}=\text{prox}_{\mathbf{p}_k}(\gamma_k\nabla Z(\mathbf{p}_k))=\text{argmin}_{\mathbf{p}\in\mathscr{A}}\,\gamma_k\nabla Z(\mathbf{p}_k)'(\mathbf{p}-\mathbf{p}_k)+V(\mathbf{p}_k,\mathbf{p}).$$

This procedure is known as the entropic descent algorithm (EDA).

Since $\nabla Z(\mathbf{p}_k)$ is not known exactly in the stochastic simulation context, one needs to estimate it by sampling. In the next sections, we shall discuss two issues with implementing a stochastic version of EDA: The first is how to obtain an estimate of the gradient, and the second is on efficient methods for solving the subprogram that defines the prox-mapping.

In the convex optimization setting where unbiased samples for ∇Z is available, it is known that the convergence rate of the optimality gap, i.e. $E[Z(x_k) - Z(x^*)]$ for MDSA is of order $1/\sqrt{k}$. Compared to the projective gradient method, i.e. carrying out an Euclidean projection on the feasible region at each iteration, which is in fact a special case of MDSA by taking $\omega(x) = (1/2)\|x\|_2^2$ where $\|\cdot\|_2$ denotes L_2-norm, the convergence constant in front of $1/\sqrt{k}$ for EDA can be $\sqrt{n/\log n}$ times smaller (Nemirovski et al. 2009). This makes EDA favorable for large n, and motivates our use of EDA in this paper.

Another motivation for using MDSA is that, in contrast to FWSA proposed in Ghosh and Lam (2015), we do not increase the sample size with the iteration number. To explain the difference in the required sample sizes on an intuitive level, we note that FWSA consists of solving a subprogram at each iteration with a linearized objective in order to obtain the best feasible direction. This subprogram introduces bias in estimating the best feasible direction, even when the gradient itself can be estimated unbiasedly. The increase in sample size with respect to iteration is therefore designed to offset this bias. On the other hand, prox-mapping possesses contraction properties similar to a projection, which facilitates the analysis for MDSA using a notion of distance derived from the prox-function rather than by looking at the bias separately.

3 GRADIENT ESTIMATION

Note that since $\mathbf{p} \in \mathscr{P}$ is a discrete distribution, we can write

$$Z(\mathbf{p}) = E_{\mathbf{p}}[h(\mathbf{X})] = \sum_{i_1} \cdots \sum_{i_T} h(x_{i_1}, \ldots, x_{i_T}) p_{i_1} \cdots p_{i_T} \tag{5}$$

so that $Z(\mathbf{p})$ is a high-dimensional polynomial in the variables $\mathbf{p}$. Although explicit form (5) is available, there are often numerous summands in (5) that renders exact computation infeasible, and so simulation has to be used. In carrying out gradient estimation for (5), one issue is that naive differentiation of (5) with respect to each coordinate of $\mathbf{p}$ does not lead to any implementable procedure, because it results in a high-dimensional multivariate polynomial that has no probabilistic meaning.

To get around the above issue, Ghosh and Lam (2015) proposes using the Gateaux derivative on a functional of probability distribution for simulating the gradient of (5), and we shall use it here. Given a $\mathbf{p}$, consider a mixture of $\mathbf{p}$ with the point mass at each coordinate, i.e. $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0)'$ where the 1 is at the i-th coordinate. This mixture is written as $(1-\varepsilon)\mathbf{p} + \varepsilon \mathbf{e}_i$ for $i = 1, \ldots, n$. Then consider

$$\psi_i(\mathbf{p}) = \frac{d}{d\varepsilon} Z((1-\varepsilon)\mathbf{p} + \varepsilon \mathbf{e}_i)\bigg|_{\varepsilon=0} \tag{6}$$

and define $\boldsymbol{\psi}(\mathbf{p}) = (\psi_1(\mathbf{p}), \ldots, \psi_n(\mathbf{p}))'$. This vector $\boldsymbol{\psi}(\mathbf{p})$ captures all the necessary gradient information of $Z(\mathbf{p})$ within the space $\mathscr{P}$, in the following sense:

Theorem 1 (Adapted from Ghosh and Lam (2015)) The vector $\boldsymbol{\psi}(\mathbf{p})$ defined in (6) satisfies

$$\nabla Z(\mathbf{p})'(\mathbf{q} - \mathbf{p}) = \boldsymbol{\psi}(\mathbf{p})'(\mathbf{q} - \mathbf{p}) \tag{7}$$

for any $\mathbf{p}, \mathbf{q} \in \mathscr{P}$, and can be evaluated by writing as

$$\psi_i(\mathbf{p}) = E_{\mathbf{p}}[h(\mathbf{X}) s_i(\mathbf{X})] \tag{8}$$

where

$$s_i(x_1, \ldots, x_T) = \sum_{t=1}^{T} \frac{I(x_t = u_i)}{p_i} - T. \tag{9}$$

Here $I(\cdot)$ is the indicator function, and $(u_1,\ldots,u_n)$ are the support points for the probability simplex $\mathscr{P}$.

Since the operations in MDSA only involve $\nabla Z(\mathbf{p})'(\mathbf{q}-\mathbf{p})$ but not the gradient itself directly, (7) dictates that one can use $\boldsymbol{\psi}(\mathbf{p})$ to replace $\nabla Z(\mathbf{p})$. The functions s_i in (9) can be regarded as a nonparametric version of the score function (Ghosh and Lam 2015), and with (8) one can use $h(\mathbf{X})s_i(\mathbf{X})$ as an unbiased estimator for the gradient information vector $\boldsymbol{\psi}(\mathbf{p})$. In summary, given $\mathbf{p}_k$, our MDSA computes at iteration k

$$\mathbf{p}_{k+1} = \text{prox}_{\mathbf{p}_k}(\gamma_k \hat{\boldsymbol{\psi}}(\mathbf{p}_k))$$

where

$$\hat{\boldsymbol{\psi}}(\mathbf{p}_k) = \frac{1}{M}\sum_{j=1}^{M}(h(\mathbf{X})s_i(\mathbf{X}))_j$$

and $(h(\mathbf{X})s_i(\mathbf{X}))_j, j=1,\ldots,M$ denote the samples using M independent sample paths of $\mathbf{X}$.

4 COMPUTING PROX-MAPPING

We focus on two examples in this section.

4.1 Kullback-Leibler (KL) Divergence Constraint

Let

$$\mathscr{A} = \left\{\mathbf{p}\in\mathscr{P} : \sum_{i=1}^{n} p_i \log\frac{p_i}{b_i} \le \eta\right\} \tag{10}$$

where $\mathbf{b}=(b_1,\ldots,b_n)$ is some baseline distribution. This type of constraints has been used commonly in robust control (e.g. Hansen and Sargent 2008, Petersen et al. 2000) and mathematical finance (e.g. Glasserman and Xu 2014, Glasserman and Xu 2013), among other disciplines. The prox-mapping $\text{prox}_{\mathbf{p}}(\boldsymbol{\xi})$ for $\mathbf{p}=(p_1,\ldots,p_n)\in\mathscr{P}$ and $\boldsymbol{\xi}=(\xi_1,\ldots,\xi_n)\in\mathbb{R}^n$ is described as follows:

Proposition 1 For the feasible region defined in (10), the prox-mapping for the minimization problem in (1) using prox-function defined in (4) is given by $\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}) = ((\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_1,\ldots,(\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_n)$ as below: First define

$$\varphi(\mathbf{p},\boldsymbol{\xi}) = \frac{\sum_{i=1}^n p_i e^{-\xi_i}\left(\log\frac{p_i}{b_i}-\xi_i\right)}{\sum_{i=1}^n p_i e^{-\xi_i}} - \log\sum_{i=1}^n p_i e^{-\xi_i}$$

and

$$\kappa(\beta;\mathbf{p},\boldsymbol{\xi}) = \frac{\sum_{i=1}^n p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}\left(\log\frac{p_i}{b_i}-\xi_i\right)}{(1+\beta)\sum_{i=1}^n p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}} - \log\sum_{i=1}^n p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}.$$

If $\varphi(\mathbf{p},\boldsymbol{\xi})\le\eta$, then

$$(\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_i = \frac{p_i e^{-\xi_i}}{\sum_{j=1}^n p_j e^{-\xi_j}}$$

else if $\varphi(\mathbf{p},\boldsymbol{\xi})>\eta$, then

$$(\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_i = \frac{p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}}{\sum_{j=1}^n p_j^{\frac{1}{1+\beta}} b_j^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_j}{1+\beta}}}$$

where β is the positive root of the equation

$$\kappa(\beta;\mathbf{p},\boldsymbol{\xi}) = \eta. \tag{11}$$

Proposition 1 reveals that the prox-mapping under KL constraint is either an exponential twisting with exponent $-\xi_i$, or an exponential twisting together with a "geometric mixture" of $\mathbf{p}$ and $\mathbf{b}$ given by $p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}}$. Note that the main computation in the prox-mapping is merely a one-dimensional root-finding problem for β.

Proof of Proposition 1. Consider

$$\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}) = \text{argmin}_{\mathbf{q}\in\mathscr{A}} \boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + V(\mathbf{p},\mathbf{q})$$

where $V(\mathbf{p},\mathbf{q})$ is defined in (4), which can be rewritten as the solution of

$$\begin{array}{ll} \min & \boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} \\ \text{subject to} & \sum_{i=1}^n q_i \log \frac{q_i}{b_i} \le \eta \\ & \sum_{i=1}^n q_i = 1 \\ & q_i \ge 0 \quad \text{for all} \quad i = 1,\ldots,n. \end{array} \tag{12}$$

Relaxing the first constraint in (12), we have the Lagrangian

$$\boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} + \beta \left(\sum_{i=1}^n q_i \log \frac{q_i}{b_i} - \eta \right) \tag{13}$$

where β is the Lagrange multiplier. By Theorem 1, P.220 in Luenberger (1968), since the program (12) is convex, if we can find $\beta \ge 0$ and $\mathbf{q} \in \mathscr{A}$ such that $\mathbf{q}$ minimizes (13) over $\mathscr{P}$ for the fixed $\beta \ge 0$, and that β and $\mathbf{q}$ satisfy the complementary slackness condition $\beta \left(\sum_{i=1}^n b_i \log \frac{q_i}{b_i} - \eta \right) = 0$, then this $\mathbf{q}$ is an optimal solution for (12).

Suppose that $\beta = 0$. Then (13) reduces to

$$\boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i}. \tag{14}$$

To minimize (14) over $\mathbf{q} \in \mathscr{P}$, we consider a further relaxation of the constraint $\sum_{i=1}^n q_i = 1$ to get

$$\boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} + \lambda \left(\sum_{i=1}^n q_i - 1 \right). \tag{15}$$

Differentiate (15) with respect to each q_i and set it to zero to get

$$\xi_i + \log q_i + 1 - \log p_i + \lambda = 0$$

or

$$q_i \propto p_i e^{-\xi_i}.$$

The constraint $\sum_{i=1}^n q_i = 1$ implies that

$$q_i = \frac{p_i e^{-\xi_i}}{\sum_{j=1}^n p_j e^{-\xi_j}}. \tag{16}$$

It is easy to check that $\mathbf{q}$ defined by (16) satisfies the KKT condition as the minimizer of the convex minimization program of (14) over $\mathbf{q} \in \mathscr{P}$.

Now, we go back to the original problem (12). From the discussion above, $\mathbf{q}$ defined by (16) is an optimal solution for (12) if it is feasible in $\mathscr{A}$. In other words, we have to check that $\sum_{i=1}^n q_i \log \frac{q_i}{b_i} \le \eta$.

Substituting (16) into $\sum_{i=1}^n q_i \log \frac{q_i}{b_i}$ gives $\varphi(\mathbf{p},\boldsymbol{\xi})$, and so the condition becomes $\varphi(\mathbf{p},\boldsymbol{\xi}) \le \eta$. This gives rise to the first case in Proposition 1.

Suppose that $\varphi(\mathbf{p},\boldsymbol{\xi}) > \eta$. Then the $\mathbf{q}$ considered in (16) is not feasible, and we must choose $\beta > 0$. Relaxing the constraint $\sum_{i=1}^n q_i = 1$ on the Lagrangian (13), we have

$$\boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} + \beta \left(\sum_{i=1}^n q_i \log \frac{q_i}{b_i} - \eta \right) + \lambda \left(\sum_{i=1}^n q_i - 1 \right). \tag{17}$$

Differentiate with respect to each q_i and set it to zero gives

$$\xi_i + \log q_i + 1 - \log p_i + \beta(\log q_i + 1 - \log b_i) + \lambda = 0$$

or

$$q_i \propto p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}.$$

The constraint $\sum_{i=1}^n q_i = 1$ gives

$$q_i = \frac{p_i^{\frac{1}{1+\beta}} b_i^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_i}{1+\beta}}}{\sum_{j=1}^n p_j^{\frac{1}{1+\beta}} b_j^{\frac{\beta}{1+\beta}} e^{-\frac{\xi_j}{1+\beta}}}. \tag{18}$$

$\mathbf{q}$ defined by (18) satisfies the KKT condition as the minimizer of (17) over $\mathbf{q} \in \mathscr{P}$. Going back to the original problem (12), this $\mathbf{q}$ is an optimal solution if β is chosen such that $\sum_{i=1}^n q_i \log \frac{q_i}{b_i} = \eta$. Substituting (18) into $\sum_{i=1}^n q_i \log \frac{q_i}{b_i}$ gives $\kappa(\beta;\mathbf{p},\boldsymbol{\xi})$ and so $\sum_{i=1}^n q_i \log \frac{q_i}{b_i} = \eta$ is equivalent to (11). Note that we can always find a positive root for (11) in this case because when $\beta = 0$,

$$\kappa(0;\mathbf{p},\boldsymbol{\xi}) = \varphi(\mathbf{p},\boldsymbol{\xi}) > \eta$$

while

$$\kappa(\beta;\mathbf{p},\boldsymbol{\xi}) \to 0$$

as $\beta \to \infty$. Hence there must be a positive root for β since $\kappa(\beta;\mathbf{p},\boldsymbol{\xi})$ is continuous in β. □

4.2 Moment Constraint

Let

$$\mathscr{A} = \left\{ \mathbf{p} \in \mathscr{P} : \sum_{i=1}^n p_i r_l(x_i) \le \mu_l, l = 1,\dots,m \right\} \tag{19}$$

where $r_l(\cdot)$'s are some basis functions. For instance, $r_1(x) = x$ and $r_2(x) = x^2$ correspond to the first two moments. The prox-mapping $\text{prox}_{\mathbf{p}}(\boldsymbol{\xi})$ is described as follows:

Proposition 2 For the feasible region defined in (19), the prox-mapping for the minimization problem in (1) using the prox-function defined in (4) is given by $\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}) = ((\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_1,\dots,(\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_n)$, where

$$(\text{prox}_{\mathbf{p}}(\boldsymbol{\xi}))_i = \frac{p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)}}{\sum_{j=1}^n p_j e^{-\xi_j - \sum_{l=1}^m \beta_l r_l(x_j)}}$$

and $\{\beta_l, l = 1,\dots,m\}$ is the minimizer of

$$\min_{\beta_l \ge 0, l=1,\dots,m} \log \sum_{i=1}^n p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)} + \sum_{l=1}^m \beta_l \mu_l. \tag{20}$$

The prox-mapping applies an exponential twisting on the probability distribution, with exponent $-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)$. Each application of prox-mapping requires solving a convex optimization with only non-negativity constraints in a dimension equal to the number of moment constraints. If the inequalities are replaced by equalities, the procedure is exactly the same except that the corresponding Lagrange multipliers β_l in (20) become unconstrained.

Proof of Proposition 2. The prox-mapping $\text{prox}_{\mathbf{p}}(\boldsymbol{\xi})$ is found by solving

$$\begin{array}{ll} \min & \boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} \\ \text{subject to} & \sum_{i=1}^n q_i r_l(x_i) \le \mu_l, \quad l = 1,\ldots,m \\ & \sum_{i=1}^n q_i = 1 \\ & q_i \ge 0 \quad \text{for all} \quad i = 1,\ldots,n. \end{array} \tag{21}$$

Relaxing the first constraint, (21) has the same optimal value as the Lagrangian formulation

$$\max_{\beta_l \ge 0, l=1,\ldots,m} \min_{\mathbf{q}\in\mathscr{P}} \boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} + \sum_{l=1}^m \beta_l (r_l(x_i) q_i - \mu_l) \tag{22}$$

since (21) is convex. Moreover, the saddle point solution of (22) is also optimal for (21). Consider the inner minimization in (22). Relaxing further the constraint $\sum_{i=1}^n q_i = 1$, we have

$$\boldsymbol{\xi}'(\mathbf{q}-\mathbf{p}) + \sum_{i=1}^n q_i \log \frac{q_i}{p_i} + \sum_{l=1}^m \beta_l (r_l(x_i) q_i - \mu_l) + \lambda \left(\sum_{i=1}^n q_i - 1 \right).$$

Differentiate with respect to each q_i and set it to zero, we have

$$\xi_i + \log q_i + 1 - \log p_i + \sum_{l=1}^m \beta_l r_l(x_i) + \lambda = 0$$

or

$$q_i \propto p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)}.$$

The constraint $\sum_{i=1}^n q_i = 1$ implies that

$$q_i = \frac{p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)}}{\sum_{j=1}^n p_j e^{-\xi_j - \sum_{l=1}^m \beta_l r_l(x_j)}}. \tag{23}$$

$\mathbf{q}$ defined by (23) satisfies the KKT condition as the minimizer in the inner optimization in (22). Substituting (23) into the inner objective function in (22) gives

$$-\log \sum_{i=1}^n p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)} - \sum_{l=1}^m \beta_l \mu_l - \boldsymbol{\xi}'\mathbf{p}$$

so that (22) becomes

$$\max_{\beta_l \ge 0, l=1,\ldots,m} -\log \sum_{i=1}^n p_i e^{-\xi_i - \sum_{l=1}^m \beta_l r_l(x_i)} - \sum_{l=1}^m \beta_l \mu_l - \boldsymbol{\xi}'\mathbf{p}$$

whose optimal solution is the same as that of (20). □

5 NUMERICAL EXPERIMENTS

This section describes numerical experiments conducted on the canonical simulation model of the $M/GI/1$ single-server queue to characterize the performance of the MDSA algorithm. This queueing model serves as a good testing ground because a good approximation to the solution of the optimization problem (1) where the set $\mathscr{A}$ is of form (10) is available via a deterministic re-formulation, which lets us observe the rate of convergence in the objective value for our algorithm directly.

Consider an $M/GI/1$ queue where the arrival process is Poisson known with high accuracy to have rate $\lambda = 1$. On the other hand, the time X_t taken to provide service to the $t-$th customer is uncertain but is assumed i.i.d.. A simulation model estimates $Z(\mathbf{p}) = E_{\mathbf{p}}[h(\mathbf{X}_T)]$, the long-run average of the waiting times, i.e. the time between the customer's arrival and for service to commence, of the first T customers:

$$Z(\mathbf{p}) = E\left[\frac{1}{T}\sum_{1}^{T} W_t\right], \qquad \text{where} \quad W_t = \max\{0, W_{t-1} + X_t - A_t\}.$$

Note that the t-th customer's waiting time W_t is calculated by iterating over the celebrated Lindley's recursion, and $A_t \sim exp(1)$ is the inter-arrival time of the $(t-1)$-th customer.

We carry out optimization (1) that captures the worst-case performance for $Z(\mathbf{p})$ amongst service time distributions $\mathbf{p}$ that is η distance away from b in terms of KL divergence (focusing on the minimization formulation):

$$\min_{\mathbf{p}\in\mathscr{P}} Z(\mathbf{p}) \quad \text{s.t.} \quad \sum_i p_i \log\left(\frac{p_i}{b_i}\right) \leq \eta \tag{24}$$

where $\mathscr{P}$ denotes the probability simplex on $\{u_1, \ldots, u_n\}$. Here we choose u_i's as the uniform discretization of the interval $[0,1]$, i.e. $u_i = (i+1)/n$, and the nominal distribution $\mathbf{b}$ is the corresponding discretization of a mixture distribution $0.3 \times \text{Beta}(2,6) \; + 0.7 \times \text{Beta}(6,2)$ over the interval $[0,1]$, i.e. b_i is the probability mass of this continuous distribution assigned to the interval $[u_{i-1}, u_i]$.

As T grows, the average waiting time converges to the corresponding steady-state value, which is given in closed form for $M/GI/1$ queues when the traffic intensity $\rho_{\mathbf{p}} = E_{\mathbf{p}}[X_t]$ is less than 1, by the Pollaczek-Khinchine formula (Khintchine 1932) as:

$$Z_\infty(\mathbf{p}) = \frac{\rho_{\mathbf{p}} E_{\mathbf{p}}[X_1] + Var_{\mathbf{p}}(X_1)}{2(1-\rho_{\mathbf{p}})}.$$

So, when T is large, an approximation Z^*_∞ to the worst-case performance estimate can be obtained by replacing $Z(\mathbf{p})$ in the problem (24) with $Z_\infty(\mathbf{p})$. (In experiments, a choice of $T = 2000$ seems to show close agreement.) With $E_{\mathbf{p}}[X_1] = \sum p_i x_i$ and $E_{\mathbf{p}}[X_1^2] = \sum p_i x_i^2$, an approximation for the minimization problem (24) is given by the steady-state formulation **(SS)** (shown below) which is in the form of a so-called convex-concave-fractional program. As Ghosh and Lam (2015) has noted (also see Boyd and Vandenberghe 2009), **(SS)** can be reformulated as an equivalent convex optimization problem **(SS′)** using the variable substitutions $t = 1/(2(1-\sum p_i x_i))$ and $y_i = p_i t$. An approximation for the worst-case maximization problem can be similarly derived.

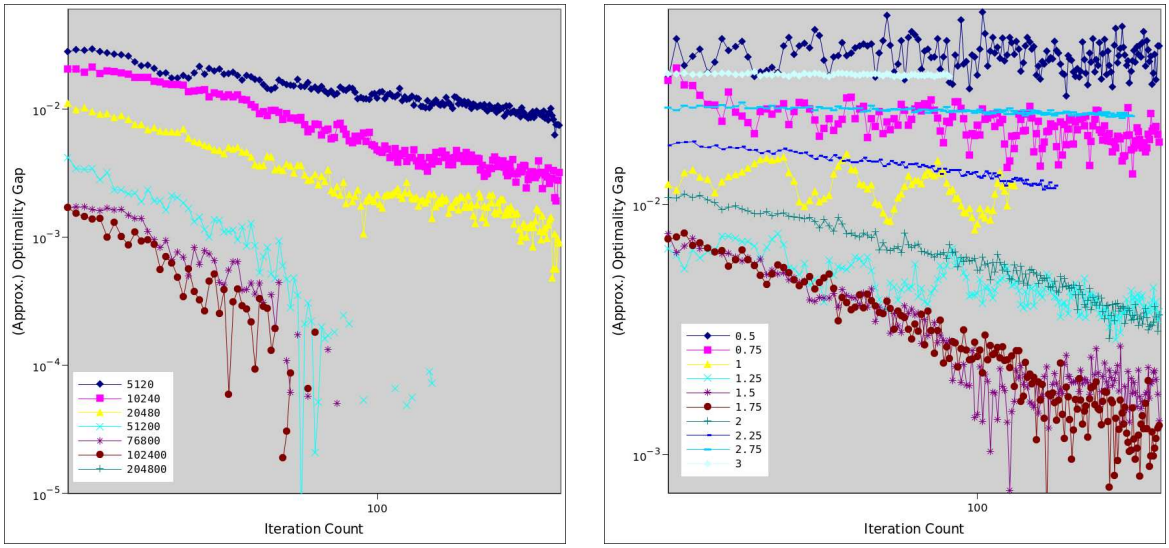
$$\begin{aligned} \min_{\mathbf{p}} \quad & \frac{\sum p_i x_i^2}{2(1-\sum p_i x_i)} && \textbf{(SS)} \\ s.t. \quad & \sum_i p_i \log\left(\frac{p_i}{b_i}\right) \leq \eta \\ & \sum p_i = 1 \\ & 0 \leq p_i \leq 1, \quad \forall i = 1, \ldots, n \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{p}} \quad & \sum y_i x_i^2 && \textbf{(SS′)} \\ s.t. \quad & \sum y_i \log\left(\frac{y_i}{t b_i}\right) \leq \eta t \\ & 2t - 2\sum y_i x_i = 1 \\ & \sum y_i = t \\ & 0 \leq y_i \leq t \quad \forall i = 1, \ldots, n \end{aligned}$$

The key parameters in the MDSA algorithm are the fixed sample-size M and the step-size sequence γ_k. We take the latter to be of the form $\gamma_k = k^{-\alpha}$. We shall experiment on the convergence behavior of the algorithm for varied M and α. Note that Nemirovski et al. (2009) study the case $\alpha = 0.5$ in particular.

All the experimental results were terminated for the MDSA algorithm at iteration k when at least one of the following criteria are met:

1. The relative difference between objective value $Z(\mathbf{p}_k)$ and the average of the observed values in 30 previous iterations, $(\sum_{v=1}^{30} Z(\mathbf{p}_{k-v}))/30$, is below 5×10^{-6}, or
2. The gradient estimate $\hat{\psi}(\mathbf{p}_k)$ has l_2-norm smaller than 1×10^{-3}, or
3. The change in iterates $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$ has l_1-norm smaller than 1×10^{-6}, or
4. The iteration count k reaches 200.



(a) $\alpha = 1.5$, M varied as shown

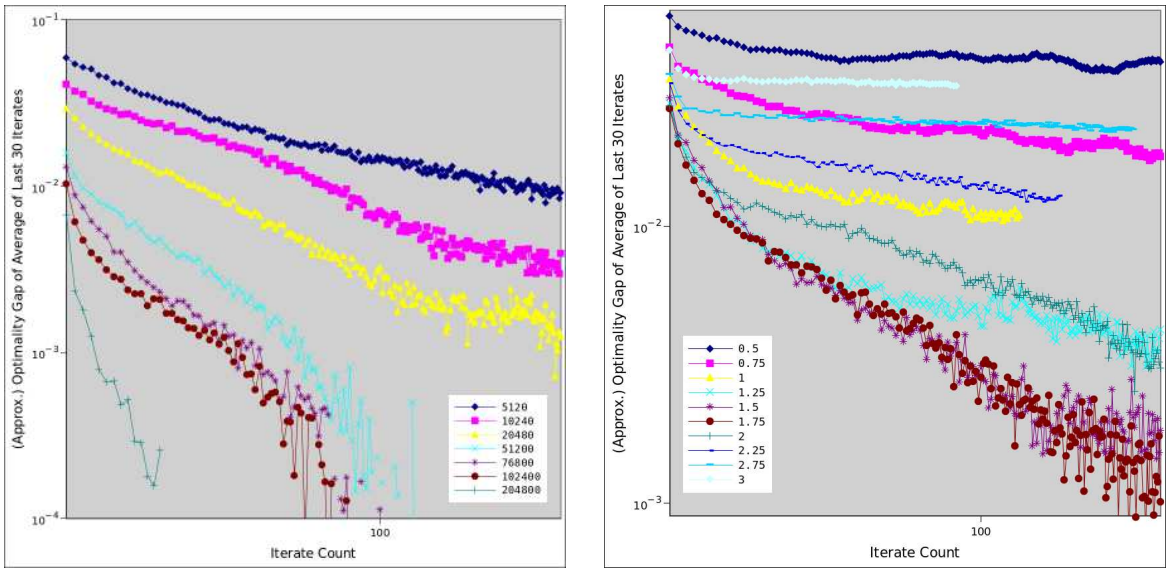
(b) $M = 76,800$, α varied as shown

Figure 1: Approximated Optimality gap of the MDSA algorithm iterates as function of iteration count (both in log-scale). On the left, the step-size exponent α is held constant at 1.5 while the sample size M is varied. Conversely, the plot on the right holds the sample-size M at 76,800 while the step-size exponent α is varied.

Fig. 1 captures the performance of the MDSA algorithm as a function of the sample size M and step-size exponent α parameters. It plots the (approximate) optimality gap as a function of the iteration count of the algorithm, where the parameters are set to be $\eta = 0.025$, $n = 100$ and $T = 2000$. As can be expected, Fig. 1a shows that the performance of the MDSA algorithm generally improves as the the sample size M used to evaluate the gradient $\psi(\mathbf{p})$ is increased. Note that Fig. 1a is plotted in log-scale, and the relatively constant slopes among different M's show that M seems to have primarily an effect on the constant factor that accompanies the first order rate of convergence of the MDSA method as a function of the iteration count. Note that for very large M values, the numerical accuracy issues intervene, resulting in the breaks observed in the data series.

Fig. 1b plots the performance as the step-size exponent α is varied. Compared to M, α affects a bit more the slopes of the trends (although still not very significantly), which demonstrates some effects on the rate of convergence of MDSA in the power order of the iteration count. The result shows that the best performance is achieved when $\alpha \approx 1.5$. When α is too aggressive (say $\alpha \geq 2.0$) the step-sizes become too small which limits the movement of the solution iterates. On the other hand, too small an α (say $\alpha \leq 1.25$)

may lead to overshooting the best next iterate; this can be inferred from the noisier sample paths for small α. (The noise observed in the sample paths for $\alpha = 1.5$ or 1.75 at the lower-right end of the plot is more likely due to numerical accuracy issues in deriving the approximation from **(SS')**.)



(a) $\alpha = 1.5$, M varied as shown

(b) $M = 76,800$, α varied as shown

Figure 2: Approximated Optimality gap of the rolling simple average of the previous thirty MDSA algorithm iterates as function of iteration count (both in log-scale). On the left, the step-size exponent α is held constant at 1.5 while the sample size M is varied. Conversely, the plot on the right holds the sample-size M at $76,800$ while the step-size exponent α is varied.

Next, we also experiment some averaging on the MDSA scheme. This sort of averaging has been considered in Nemirovski et al. (2009), where the weights are defined as a specific function of the step-size, and a uniform version has also been widely used in the classical SA literature (e.g. Polyak 1990 and Ruppert 1991). Conceptually, averaging helps tamp down the effect of the noise induced by the fixed sample size at every iteration. Fig. 2 provides the performance of a simple rolling average of the previous thirty iterates. We see that the trends noticed in Fig. 1 emerge more clearly in Fig. 2, with the noise of the trends being smaller across the board.

ACKNOWLEDGMENTS

The second author is grateful to the support by the National Science Foundation under Grants CMMI-1400391 and CMMI-1436247.

REFERENCES

Banks, J., J. Carson, B. Nelson, and D. Nicol. 2000. *Discrete-Event System Simulation*. Prentice hall Englewood Cliffs, NJ, USA.

Barton, R. R., B. L. Nelson, and W. Xie. 2013. "Quantifying Input Uncertainty via Simulation Confidence Intervals". *INFORMS Journal on Computing* 26 (1): 74–87.

Beck, A., and M. Teboulle. 2003. "Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization". *Operations Research Letters* 31 (3): 167–175.

Ben-Tal, A., D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. 2013. "Robust Solutions of Optimization Problems Affected by Uncertain Probabilities". *Management Science* 59 (2): 341–357.

Boyd, S., and L. Vandenberghe. 2009. *Convex Optimization*. Cambridge university press.
Chick, S. E. 2001. "Input Distribution Selection for Simulation Experiments: Accounting for Input Uncertainty". *Operations Research* 49 (5): 744–758.
Delage, E., and Y. Ye. 2010. "Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems". *Operations Research* 58 (3): 595–612.
Ghosh, S., and H. Lam. 2015. "Computing Worst-Case Input Models in Stochastic Simulation". *submitted, available at: http://www-personal.umich.edu/ khlam/files/robustSAOR3.pdf*.
Glasserman, P., and X. Xu. 2013. "Robust Portfolio Control with Stochastic Factor Dynamics". *Operations Research* 61 (4): 874–893.
Glasserman, P., and X. Xu. 2014. "Robust Risk Measurement and Model Risk". *Quantitative Finance* 14 (1): 29–58.
Goh, J., and M. Sim. 2010. "Distributionally Robust Optimization and its Tractable Approximations". *Operations Research* 58 (4-Part-1): 902–917.
Hansen, L. P., and T. J. Sargent. 2008. *Robustness*. Princeton university press.
Khintchine, A. Y. 1932. "Mathematical Theory of a Stationary Queue". *Matematicheskii Sbornik* 39:7384.
Luenberger, D. G. 1968. *Optimization by Vector Space Methods*. John Wiley & Sons.
Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro. 2009. "Robust Stochastic Approximation Approach to Stochastic Programming". *SIAM Journal on Optimization* 19 (4): 1574–1609.
Nemirovski, A., and D. Yudin. 1983. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York.
Petersen, I. R., M. R. James, and P. Dupuis. 2000. "Minimax Optimal Control of Stochastic Uncertain Systems with Relative Entropy Constraints". *IEEE Transactions on Automatic Control* 45 (3): 398–412.
Polyak, B. T. 1990. "New Stochastic Approximation Type Procedures". *Automat. Remote Control* 7:98107.
Ruppert, D. 1991. "Stochastic Approximation". In *Handbook in Sequential Analysis*, edited by B. K. Ghosh and P. K. Sen. Marcel Dekker.

AUTHOR BIOGRAPHIES

SOUMYADIP GHOSH is a Research Staff Member in the Business Analytics and Mathematical Sciences Department at the IBM T.J. Watson Research Center. His current research interests lie in simulation based optimization techniques for stochastic optimization problems, with a focus on applications in Energy and Power systems and supply chain management. His email is ghoshs@us.ibm.com and his web page is at https://researcher.ibm.com/researcher/view.php?person=us-ghoshs.

HENRY LAM is an Assistant Professor in the Department of Industrial and Operations Engineering at the University of Michigan, Ann Arbor. He graduated from Harvard University with a Ph.D. degree in statistics in 2011, and has been an Assistant Professor in the Department of Mathematics and Statistics at Boston University until 2014. His research focuses on stochastic simulation, risk analysis, and simulation optimization. His email address is khlam@umich.edu.