Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SIMULATION OPTIMIZATION VIA GRADIENT-BASED STOCHASTIC SEARCH

Enlu Zhou

H. Milton Stewart School of
Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Shalabh Bhatnagar

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012, INDIA

Xi Chen

Department of Industrial & Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

ABSTRACT

Based on model-based methods, a recent class of stochastic search methods for nonlinear deterministic optimization, we propose a new algorithm for simulation optimization over continuous space. The idea is to reformulate the original simulation optimization problem into another optimization problem over the parameter space of the sampling distribution in model-based methods, and then use a direct gradient search on the parameter space to update the sampling distribution. To improve the computational efficiency, we further develop a two-timescale updating scheme that updates the parameter on a slow timescale and estimates the quantities involved in the parameter updating on a fast timescale. We provide numerical experiments to illustrate the performance of our algorithms.

1 INTRODUCTION

We consider simulation optimization over continuous space, where the objective function cannot by evaluated exactly but by a simulation approximation. As characterized by Fu, Chen, and Shi (2008), there are four main classes of approaches to simulation optimization over continuous space: (i) sample average approximation, e.g. de Mello, Shapiro, and Spearman (1999); (ii) stochastic gradient methods or stochastic approximation, e.g. Kiefer and Wolfowitz (1952), Kushner and Yin (2004); (iii) sequential response surface methodology, e.g. Barton and Meckesheimer (2006), Chang, Hong, and Wan (2013); and (iv) deterministic metaheuristics, a broad category of methods that generalize deterministic metaheuristics to the simulation optimization setting, e.g. Olafsson (2006), Andradóttir (2006).

Among deterministic metahuristics, a recent class of algorithms under the name of "model-based methods" have shown good performance in a wide range of applications. These methods include ant colony optimization (Dorigo and Gambardella 1997), annealing adaptive search (AAS) (Romeijn and Smith 1994), probability collectives (PCs) (Wolpert 2004), the estimation of distribution algorithms (EDAs) (Larranaga et al. 1999), the cross-entropy (CE) method (DeBoer et al. 2005), model reference adaptive search (MRAS) (Hu, Fu, and Marcus 2007), the interacting-particle algorithm (Molvalioglu, Zabinsky, and Kohn 2009), and the gradient-based adaptive stochastic search (GASS) (Zhou and Hu 2012, Zhou and Hu 2014). Model-based methods typically assume a sampling distribution (i.e., a probabilistic model), often within a parameterized family of distributions, over the solution space, and iteratively carry out the two interrelated steps: (1) draw candidate solutions from the sampling distribution; (2) use the evaluations of these candidate solutions to

update the sampling distribution. The hope is that at every iteration the sampling distribution is biased towards the more promising regions of the solution space, and will eventually concentrate on one or more of the optimal solutions.

A straightforward idea for extending model-based methods to simulation optimization is to simulate each candidate solution at a given iteration for an equal number of times to obtain their performance estimates and then use these estimates to update the probability distribution over the solution space. To improve computational efficiency, better simulation allocation rules have been proposed to replace the equal allocation rule. Hu, Fu, and Marcus (2008) studied the sufficient conditions on the simulation allocation rule in MRAS to guarantee convergence of the algorithm. Chepuri and de Mello (2005) proposed a simple heuristic sampling scheme to determine the number of simulation replications in each iteration for the CE method. More recently, He et al. (2010) developed the algorithm under the name of cross-entropy with optimal computing budget allocation (CEOCBA), which nicely incorporates the idea of optimal computing budget allocation (OCBA) (Chen et al. 2000) into each individual iteration of the CE method.

Unlike the aforementioned approaches, we propose a different idea of adapting model-based methods to simulation optimization: we reformulate the original problem as a new optimization problem over the parameter space of the sampling distribution, and then use a direct gradient search to update the parameter. This idea is an extension of the gradient-based adaptive stochastic search (GASS) algorithm that we developed earlier for deterministic non-differentiable optimization (cf. Zhou and Hu (2012), Zhou and Hu (2014) for details on GASS). In the current simulation optimization setting, the gradient and Hessian terms involved in the gradient search need to be estimated jointly by candidate solutions and their performance evaluations. Aiming at reducing the sample size to further improve the efficiency, we also develop a two-timescale scheme that updates the parameter on a slow timescale and estimates the gradient and Hessian terms using the samples on a fast timescale. The resultant algorithm is essentially a two-timescale stochastic approximation scheme (cf. Borkar (2008), Bhatnagar and Borkar (1998), Bhatnagar, Prasad, and Prashanth (2013) for details of multi-timescale stochastic approximation).

The rest of the paper is organized as follows. Section 2 introduces the main idea. Section 3 presents our proposed algorithm and the two-timescale variant. Section 4 illustrates the performance of our algorithms by comparing with the CEOCBA method on several benchmark problems. Finally we conclude the paper in Section 5.

2 MAIN IDEA

Consider the simulation optimization problem

$$\max_{x\in\mathscr{X}} \quad H(x) = E_{\xi}[h(x,\xi)], \tag{1}$$

where $\mathscr{X} \subseteq \mathscr{R}^x$, and ξ represents the randomness in the system and follows a distribution $p(\xi)$ that is not dependent on x. Assume that H is bounded on $\mathscr{X}$, i.e., $\exists H_{lb} > -\infty$, $H_{ub} < \infty$ s.t. $H_{lb} \leq H(x) \leq H_{ub}$. Due to the complexity of system, the analytical form of H is often not available. Hence, we assume for a given x, its performance can only be evaluated by simulation or experimentation, which returns a noisy function evaluation of $H(x)$ or in other words, a sample of $h(x,\xi)$. We also assume the following condition on the objective function:

Assumption 1 For an arbitrary and fixed value z, $h(x_1,z) \geq h(x_2,z)$ if $H(x_1) \geq H(x_2)$.

This assumption is satisfied by many objective functions, such as those with an additive noise, i.e. $h(x,\xi) = \tilde{h}(x) + \xi$.

As in many model-based methods, we introduce a parameterized family of densities $\{f(x;\theta), \theta \in \Theta \subset \mathscr{R}^{\theta}\}$ as the sampling distribution, where θ is the parameter that will be updated over iterations. We will illustrate our idea by first considering a simple reformulation

$$L(\theta) \triangleq \int H(x) f(x;\theta) dx = \int\int h(x,\xi) p(\xi) f(x;\theta) dx d\xi.$$

It is easy to see that $L(\theta) \leq H(x^*)$ and the equality is achieved if and only if all the probability mass of $f(x;\theta)$ concentrates on a subset of the set of global optima. Given the existence of such a θ, we can solve the new optimization problem $\max_{\theta\in\Theta} L(\theta)$ instead of the original problem, since the optimal parameter will recover the optimal solution and the optimal function value.

For a full-blown development of our proposed algorithm, we introduce a shape function $S_\theta : \mathbb{R} \to \mathbb{R}^+$, where the subscript θ signifies the possible dependence of the shape function on the parameter θ. The function S_θ satisfies the following conditions:

(a) For every θ, $S_\theta(y)$ is increasing in y and bounded from above and below for finite y. Moreover, for every fixed y, $S_\theta(y)$ is continuous in θ.

The purpose of the shape function is to make the objective function value positive, and the choice of the shape function also helps to adjust the balance between exploitation around the current promising solutions and exploration over the entire solution space. Then for an arbitrary but fixed $\theta' \in \Theta$, define

$$L(\theta;\theta') = \int\int S_{\theta'}(h(x,\xi))p(\xi)f(x;\theta)dxd\xi.$$

Proposition 1 Under Assumption 1, for all $x \in \mathscr{X}$

$$L(\theta;\theta') \leq E_\xi\left[S_{\theta'}(h(x^*,\xi))\right],$$

and the equality is achieved if and only if all the probability mass of $f(x;\theta)$ is concentrated on a subset of the set of global optima.

Proof. From Assumption 1 and the monotonicity of the function S, it is easy to see that for any fixed value of ξ,

$$S_{\theta'}(h(x,\xi)) \leq S_{\theta'}(h(x^*,\xi)).$$

By taking the expectation with respect to ξ and x on both sides, the inequality is proved. This ends the proof.

Thus, in order to find the optimal sampling parameter we can consider

$$\max_{\theta\in\Theta} \quad l(\theta;\theta') \triangleq \ln L(\theta;\theta').$$

Since $\ln(\cdot)$ is a strictly increasing function, $l(\theta;\theta')$ has the same set of optimal solutions as $L(\theta;\theta')$. Motivated by this reformulation, we propose a stochastic search algorithm that carries out the following two steps at each iteration: let θ_k be the parameter obtained at the k^{th} iteration,

1. Generate candidate solutions from $f(x;\theta_k)$.
2. Update the parameter to θ_{k+1} using a Newton-like iteration for $\max_\theta l(\theta;\theta_k)$.

The second step above requires one to compute the gradient and Hessian of $l(\theta;\theta_k)$. For this purpose, we choose $\{f(x;\theta)\}$ to be an exponential family of densities defined as below, and the corresponding analytical expressions of the gradient and Hessian are provided in Proposition 2 that follows.

Definition 1 A family $\{f(x;\theta) : \theta \in \Theta\}$ is an exponential family of densities if it satisfies

$$f(x;\theta) = \exp\{\theta^T T(x) - \phi(\theta)\}, \quad \phi(\theta) = \ln\left\{\int \exp(\theta^T T(x))dx\right\}. \tag{2}$$

where $T(x) = [T_1(x), T_2(x), \ldots, T_d(x)]^T$ is the vector of sufficient statistics, $\theta = [\theta_1, \theta_2, \ldots, \theta_d]^T$ is the vector of natural parameters, and $\Theta = \{\theta \in \mathbb{R}^d : |\phi(\theta)| < \infty\}$ is the natural parameter space with a nonempty interior.

Proposition 2 Assume that $f(x;\theta)$ is twice differentiable on Θ and that $\nabla_\theta f(x;\theta)$ and $\nabla^2_\theta f(x;\theta)$ are both bounded on $\mathscr{X}$ for any $\theta \in \Theta$. Then

$$\begin{aligned}\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} &= E_{g(\cdot;\theta')}[\nabla_\theta \ln f(X;\theta')] \\ \nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} &= E_{g(\cdot;\theta')}[\nabla^2_\theta \ln f(X;\theta')] + \mathrm{Var}_{g(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right],\end{aligned}$$

where $E_{g(\cdot;\theta')}$ and $\mathrm{Var}_{g(\cdot;\theta')}$ denote the expectation and variance taken with respect to the probability density function (p.d.f.) defined as

$$g(x,\xi;\theta) = \frac{S_\theta(h(x,\xi))p(\xi)f(x;\theta)}{\int\int S_\theta(h(x,\xi))p(\xi)f(x;\theta)d\xi dx}.$$

Furthermore, if $f(x;\theta)$ is in an exponential family of densities, then the above expressions reduce to

$$\begin{aligned}\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} &= E_{g(\cdot;\theta')}[T(X)] - E_{\theta'}[T(X)], \\ \nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} &= \mathrm{Var}_{g(\cdot;\theta')}[T(X)] - \mathrm{Var}_{\theta'}[T(X)],\end{aligned}$$

where $E_{\theta'}$ and $\mathrm{Var}_{\theta'}$ denote the expectation and variance with respect to $f(\cdot;\theta')$.

Proof. First, consider the gradient of $l(\theta;\theta')$ with respect to θ. Notice that $\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} = \frac{\nabla_\theta L(\theta;\theta')}{L(\theta;\theta')}|_{\theta=\theta'}$ and

$$\nabla_\theta L(\theta;\theta') = \int\int S_{\theta'}(h(x,\xi))p(\xi)\nabla_\theta f(x;\theta)dxd\xi = \int\int S_{\theta'}(h(x,\xi))p(\xi)f(x;\theta)\nabla_\theta \ln f(x;\theta)dxd\xi,$$

where the interchange of integral and derivative in the first equality follows from the boundedness assumptions on $S_{\theta'}$ and $\nabla_\theta f(x;\theta)$ and the dominated convergence theorem. Then we have

$$\begin{aligned}\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} &= \left.\frac{\int\int S_{\theta'}(h(x,\xi))p(\xi)f(x;\theta)\nabla_\theta \ln f(x;\theta)dxd\xi}{L(\theta;\theta')}\right|_{\theta=\theta'} \\ &= E_{g(\cdot;\theta')}[\nabla_\theta \ln f(X;\theta')],\end{aligned} \tag{3}$$

where

$$g(x,\xi;\theta') = \frac{S_{\theta'}(h(x,\xi))p(\xi)f(x;\theta')}{\int\int S_{\theta'}(h(x,\xi))p(\xi)f(x;\theta')d\xi dx}.$$

Differentiating (3) with respect to θ and using the fact $\nabla_\theta f(x;\theta) = f(x;\theta)\nabla_\theta \ln f(x;\theta)$, we obtain the Hessian

$$\begin{aligned}\nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} &= E_{g(\cdot;\theta')}[\nabla^2_\theta \ln f(X;\theta')] + E_{g(\cdot;\theta')}\left[\nabla_{\theta'} \ln f(X;\theta')(\nabla_{\theta'} \ln f(X;\theta'))^T\right] \\ &\quad - E_{g(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right] E_{g(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right]^T \\ &= E_{g(\cdot;\theta')}[\nabla^2_\theta \ln f(X;\theta')] + \mathrm{Var}_{g(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right].\end{aligned} \tag{4}$$

Furthermore, if $f(x;\theta) = \exp\{\theta^T T(x) - \phi(\theta)\}$, we have

$$\begin{aligned}\nabla_\theta \ln f(x;\theta) &= \nabla_\theta\left(\theta^T T(x) - \ln\int \exp(\theta^T T(x))dx\right) \\ &= T(x) - \frac{\int \exp(\theta^T T(x))T(x)dx}{\int \exp(\theta^T T(x))dx} \\ &= T(x) - E_\theta[T(X)].\end{aligned} \tag{5}$$

Differentiating (5) with respect to θ, we obtain

$$\begin{aligned}\nabla^2_\theta \ln f(x;\theta) &= -\frac{\int \exp(\theta^T T(x))T(x)T(x)^T dx}{\int \exp(\theta^T T(x))dx} + \frac{\int \exp(\theta^T T(x))T(x)dx \left(\int \exp(\theta^T T(x))T(x)dx\right)^T}{\left(\int \exp(\theta^T T(x))dx\right)^2} \\ &= -E_\theta[T(X)T(X)^T] + E_\theta[T(X)]E_\theta[T(X)]^T \\ &= -\mathrm{Var}_\theta[T(X)]. \end{aligned} \tag{6}$$

Plugging (5) into (3) yields

$$\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} = E_{g(\cdot;\theta')}[T(X)] - E_{\theta'}[T(X)].$$

Plugging (5) and (6) into (4) yields

$$\nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} = \mathrm{Var}_{g(\cdot;\theta')}[T(X)] - \mathrm{Var}_{\theta'}[T(X)].$$

Thus, Proposition 2 is proved.

Noticing that the Hessian $\nabla^2_\theta l(\theta';\theta')$ is not necessarily negative definite to ensure the parameter updating is along the ascent direction of $l(\theta;\theta')$, we approximate the Hessian by the slightly perturbed second term in $\nabla^2_\theta l(\theta';\theta')$, i.e., $-(\mathrm{Var}_{\theta'}[T(X)] + \varepsilon I)$, which is negative definite. Then we can update the parameter as follows:

$$\begin{aligned}\theta_{k+1} &= \Pi_\Theta\left\{\theta_k + \alpha_k \left(\mathrm{Var}_{\theta_k}[T(X)] + \varepsilon I\right)^{-1} \nabla_\theta l(\theta_k;\theta_k)\right\} \\ &= \Pi_\Theta\left\{\theta_k + \alpha_k \left(\mathrm{Var}_{\theta_k}[T(X)] + \varepsilon I\right)^{-1} \left(E_{g(\cdot;\theta_k)}[T(X)] - E_{\theta_k}[T(X)]\right)\right\}, \end{aligned} \tag{7}$$

where α_k is a positive step-size, and Π_Θ denotes the projection operator that projects an iterate back onto the parameter space Θ by choosing the closest point in Θ.

To have an implementable algorithm, we still need to evaluate or estimate the expectation and variance terms in (7). The expectation term $E_{\theta_k}[T(X)]$ can be evaluated analytically in most cases. For example, if $\{f(\cdot;\theta_k)\}$ is chosen as the Gaussian family, then $E_{\theta_k}[T(X)]$ reduces to the mean and second moment of the Gaussian distribution. The variance term $\mathrm{Var}_{\theta_k}[T(X)]$ may not be directly available or could be too complicated to compute analytically, but it can be approximated by the sample variance using the candidate solutions drawn from $f(\cdot;\theta_k)$. The remaining term $E_{g(\cdot;\theta_k)}[T(X)]$ can be estimated based on the principle of importance sampling, since

$$E_{g(\cdot;\theta_k)}[T(X)] \propto \int\int S_{\theta_k}(h(x,\xi))T(x)p(\xi)f(x;\theta_k)d\xi dx.$$

Thus, we draw i.i.d. sample pairs $\{(x^i_k, \xi^i_k), i = 1,\ldots,N_k\}$ from the joint distribution $p(\xi)f(x;\theta)$, compute normalized weights of the samples by

$$w^i_k \propto S_{\theta_k}(h(x^i_k,\xi^i_k)), \quad i = 1,\ldots,N_k, \quad \sum_{i=1}^{N_k} w^i_k = 1,$$

and approximate $E_{g(\cdot;\theta_k)}[T(X)]$ by

$$\widehat{E}_{g_k}[T(X)] = \sum_{i=1}^{N_k} w^i_k T(x^i_k).$$

Note that in the estimate above, only one performance evaluation $h(x^i_k,\xi^i_k)$ is required for each candidate solution x^i_k, that is, we only need to carry out simulation or experimentation once for each candidate solution. This is different from existing approaches, such as CEOCBA (He et al. 2010), which require multiple evaluations for every candidate solution in order to obtain a performance estimate of certain accuracy. In other words, our algorithm avoids the problem of simulation budget allocation among candidate solutions.

3 ALGORITHM: GASSO

With the derivations above, we propose the following algorithm.

Algorithm 1 **G**radient-based **A**daptive **S**tochastic Search for **S**imulation **O**ptimization (GASSO)

1. *Initialization:* choose an exponential family of densities $\{f(\cdot;\theta)\}$, and specify a small positive constant ε, initial parameter θ_0, sample size sequence $\{N_k\}$ that is non-decreasing, and step size sequence $\{\alpha_k\}$ that satisfies $\sum_{k=0}^{\infty}\alpha_k=\infty, \sum_{k=0}^{\infty}\alpha_k^2<\infty$. Set $k=0$.
2. *Sampling:* draw samples $x_k^i \overset{\text{iid}}{\sim} f(x;\theta_k), i=1,2,\ldots,N_k$. For each x_k^i, evaluate the performance once to generate $h(x_k^i,\xi_k^i)$.
3. *Estimation:* compute the normalized weights w_k^i according to
$$w_k^i = \frac{S_{\theta_k}(h(x_k^i,\xi_k^i))}{\sum_{j=1}^{N_k} S_{\theta_k}(h(x_k^j,\xi_k^j))},$$
and estimate $E_{g_k}[T(X)]$ and $\mathrm{Var}_{\theta_k}[T(X)]$ as follows:
$$\begin{aligned}\widehat{E}_{g_k}[T(X)] &= \sum_{i=1}^{N_k} w_k^i T(x_k^i),\\ \widehat{\mathrm{Var}}_{\theta_k}[T(X)] &= \frac{1}{N_k-1}\sum_{i=1}^{N_k} T(x_k^i)T(x_k^i)^T - \frac{1}{N_k^2-N_k}\left(\sum_{i=1}^{N_k} T(x_k^i)\right)\left(\sum_{i=1}^{N_k} T(x_k^i)\right)^T.\end{aligned}$$
4. *Updating:* update the parameter θ according to
$$\theta_{k+1} = \Pi_{\tilde{\Theta}}\left\{\theta_k + \alpha_k(\widehat{\mathrm{Var}}_{\theta_k}[T(X)]+\varepsilon I)^{-1}(\widehat{E}_{g_k}[T(X)]-E_{\theta_k}[T(X)])\right\}, \tag{8}$$
where $\tilde{\Theta}\subseteq\Theta$ is a non-empty compact and convex constraint set, and $\Pi_{\tilde{\Theta}}$ denotes the projection operator that projects an iterate back onto the set $\tilde{\Theta}$ by choosing the closest point in $\tilde{\Theta}$.
5. *Stopping:* check if some stopping criterion is satisfied. If yes, stop and return the current best sampled solution; else, set $k:=k+1$ and go back to step 2.

In the initialization step (step 1) above, the conditions on the sample size sequence and step size sequence are imposed to guarantee the convergence. The sampling step (step 2) draws candidate solutions from the current sampling distribution and runs simulation or experimentation once for each candidate solution to obtain its performance evaluation. The estimation step (step 3) computes the gradient and Hessian estimates using the samples. One common choice of the shape function S is similar to the level/indicator function used in the CE method and MRAS:
$$S_\theta(h(x,\xi)) = \frac{1}{1+e^{-S_0(h(x,\xi)-\gamma_\theta)}}, \tag{9}$$
where S_0 is a large positive constant, and γ_θ is the $(1-\rho)$-quantile
$$\gamma_\theta \triangleq \sup_r\{r : P_\theta\{x\in\mathscr{X} : h(x,\xi)\geq r\}\geq\rho\},$$
where P_θ denotes the probability with respect to $f(\cdot;\theta)$. Noticing that $1/(1+e^{-S_0(h(x,\xi)-\gamma_\theta)})$ is a continuous approximation of the indicator function $I\{h(x,\xi)\geq\gamma_\theta\}$, this shape function essentially prunes the level sets below γ_θ. By varying ρ, we can adjust the percentile of elite samples that are selected to update the next sampling distribution: the smaller ρ, the less elite samples selected and hence more emphasis is put on exploiting the neighborhood of the current best solutions. The projection operator in the updating step (step 4) is primarily used to ensure the numerical stability of the algorithm. It prevents the iterates of the algorithm from becoming too big in practice and ensures the sequence $\{\theta_k\}$ to stay bounded as the search proceeds. Intuitively, such a constraint set should be chosen sufficiently large in practice so that the limits of the recursion at step 4 without the projection are contained in its interior.

3.1 Two-timescale GASSO

The parameter updating step in GASSO can be interpreted as a stochastic approximation scheme for evaluating the zeros of $\Pi_{\tilde{\Theta}}\{\mathrm{Var}_\theta[T(X)]+\varepsilon I)^{-1}\nabla_\theta l(\theta;\theta)\}=0$. To reduce the sample size per iteration and further improve the efficiency, we can also use a stochastic approximation scheme to update the Hessian and gradient estimates on a faster timescale. That is, at k^{th} iteration we carry out the following steps:

- Draw candidate solutions $x_k^i \overset{\text{iid}}{\sim} f(x;\theta_k),\ i=1,\ldots,N$, and simulate to obtain $h(x_k^i,\xi_k^i),\ i=1,\ldots,N$.
- Update the gradient and Hessian estimates on the fast timescale with step size β_k.
- Update the parameter θ_k on the slow timescale with step size α_k.

The step sizes satisfy the standard conditions in two-timescale stochastic approximation (cf. Borkar (2008)) as shown in the following assumption.

Assumption 2 The step sizes satisfy the following conditions

$$\sum_k \alpha_k = \sum_k \beta_k = \infty, \tag{10}$$

$$\sum_k (\alpha_k^2+\beta_k^2) < \infty, \tag{11}$$

$$\lim_{k\to\infty}\frac{\alpha_k}{\beta_k} = 0. \tag{12}$$

By Assumption 2, as k goes to infinity, β_k dominates α_k. So the parameter θ is updated on the slow timescale with step size α_k, while the gradient and Hessian estimates are updated on the fast timescale with step size β_k. When k is large, the parameter θ_k remains almost unchanged compared with the gradient and Hessian estimates, so θ_k is often said to be quasi-static to the fast timescale. As a result, the sampling distribution can be viewed as fixed over many iterations while the gradient and Hessian estimates are updated, which needs only a small sample size N for every iteration.

With the above idea, we propose the following two-timescale variant of GASSO.

Algorithm 2 Two-timescale **G**radient-based **A**daptive **S**tochastic Search for **S**imulation **O**ptimization (GASSO-2T)

1. *Initialization:* choose an exponential family of densities $\{f(\cdot;\theta)\}$, and specify a small positive constant ε, initial parameter θ_0, sample size N ($N>1$), and step size sequences $\{\alpha_k\}$ and $\{\beta_k\}$ that satisfy Assumption 2. Set $k=0$. Set $L_0(1)=0$, $ET_0(1)=0$, $T_0(1)=0$, $Q_0(1)=0$.
2. *Sampling:* draw samples $x_k^i \overset{\text{iid}}{\sim} f(x;\theta_k), i=1,2,\ldots,N$. For each x_k^i, evaluate the performance once to generate $h(x_k^i,\xi_k^i)$.
3. *Estimation:*
   (a) We first estimate $L(\theta;\theta_k)$ in the following manner: For $i=1,\ldots,N$, update
   $$L_k(i+1) = L_k(i)+\beta_k(S_{\theta_k}(h(x_k^i,\xi_k^i))-L_k(i)).$$
   Set $L_{k+1}(1) := L_k(N+1)$.
   (b) Next, we estimate $E_{g_k}[T(X)]$ as follows: For $i=1,\ldots,N$, update
   $$G_k(i+1) = G_k(i)+\beta_k\left(\frac{S_{\theta_k}(h(x_k^i,\xi_k^i))}{L_k(N+1)}T(x_k^i)-G_k(i)\right).$$
   Set $\widehat{E}_{g_k}[T(X)] := G_k(N+1)$. Set $G_{k+1}(1) := G_k(N+1)$.
   (c) Finally, we estimate $Var_{\theta_k}[T(X)]$ as follows: For $i=1,\ldots,N$, update
   $$\begin{aligned} P_k(i+1) &= P_k(i)+\beta_k(P(x_k^i)-P_k(i)) \\ Q_k(i+1) &= Q_k(i)+\beta_k(P(x_k^i)P(x_k^i)'-Q_k(i)). \end{aligned}$$
   Set $\widehat{\mathrm{Var}}_{\theta_k}[T(X)] := Q_k(N+1)-P_k(N+1)P_k(N+1)'$. Set $P_{k+1}(1) := P_k(N+1)$.

4. *Updating:* update the parameter θ according to

$$\theta_{k+1} = \Pi_{\tilde{\Theta}} \left\{ \theta_k + \alpha_k (\widehat{\mathrm{Var}}_{\theta_k}[T(X)] + \varepsilon I)^{-1} (\widehat{E}_{g_k}[T(X)] - E_{\theta_k}[T(X)]) \right\},$$

   where $\tilde{\Theta} \subseteq \Theta$ is a non-empty compact and convex constraint set, and $\Pi_{\tilde{\Theta}}$ denotes the projection operator that projects an iterate back onto the set $\tilde{\Theta}$ by choosing the closest point in $\tilde{\Theta}$.
5. *Stopping:* check if some stopping criterion is satisfied. If yes, stop and return the current best sampled solution; else, set $k := k+1$ and go back to step 2.

GASSO-2T differs from GASSO in the estimation step (step 3), where the expectation and variance terms are estimated through stochastic approximation iterations instead of sample average or importance sampling estimates. It is worth mentioning that GASSO and GASSO-2T can be interpreted as stochastic approximation and two-timescale stochastic approximation, respectively; therefore, it is possible to analyze the convergence properties by drawing upon techniques from stochastic approximation literature (Kushner and Yin 2004, Borkar 2008).

4 NUMERICAL EXPERIMENTS

In this section, we test GASSO and GASSO-2T on some continuous benchmark global optimization problems from Hu, Fu, and Marcus (2007) with additive noise ξ that is normally distributed with mean 0 and variance 100. We compare the performance of GASSO and GASSO-2T with the cross-entropy with optimal computing budget allocation (CEOCBA) method developed by He et al. (2010). The test functions are listed below with dimension n specified in the parentheses.

(1) Powell singular function ($n = 10$): $h_1(x,\xi) = -1 - \sum_{i=2}^{n-2} \left[(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4 \right] + \xi$, where $x^* = (0, \cdots, 0)^T$, $H_1(x^*) = E_\xi[h_1(x^*,\xi)] = -1$.

(2) Griewank function ($n = 5$): $h_2(x,\xi) = -\frac{1}{4000} \sum_{i=1}^n x_i^2 + \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1 + \xi$, where $x^* = (0, \cdots, 0)^T$, $H_2(x^*) = E_\xi[h_2(x^*,\xi)] = 0$.

(3) Trigonometric function ($n = 10$): $h_3(x,\xi) = -1 - \sum_{i=1}^n \left[8\sin^2(7(x_i - 0.9)^2) + 6\sin^2(14(x_i - 0.9)^2) + (x_i - 0.9)^2 \right] + \xi$, where $x^* = (0.9, \cdots, 0.9)^T$, $H_3(x^*) = E_\xi[h_3(x^*,\xi)] = -1$.

(4) Pintér's function ($n = 10$): $h_4(x,\xi) = -1 - \left[\sum_{i=1}^n i x_i^2 + \sum_{i=1}^n 20i \sin^2(x_{i-1}\sin x_i - x_i + \sin x_{i+1}) + \sum_{i=1}^n i \log_{10}(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2) \right] + \xi$, where $x^* = (0, \cdots, 0)^T$, $H_4(x^*) = E_\xi[h_4(x^*,\xi)] = -1$.

Specifically, Powell (H_1) is badly-scaled function; Griewank (H_2) and Trigonometric (H_3) are multimodal functions with a large number of local optima; Pintér (H_4) is both multimodal and badly-scaled.

In all these three algorithms, we use independent multivariate normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$ as the parameterized distributions $f(\cdot;\theta_k)$, where $\Sigma_k = diag(\sigma_k^2)$, $\theta_k = (\mu_k, \sigma_k^2)$, and k is the iteration number. The initial mean μ_0 is chosen randomly according to uniform distributions on $[-30, 30]^n$, and the initial covariance matrix is set to be $\Sigma_0 = 1000 I_{n\times n}$, where $I_{n\times n}$ is the identity matrix with size n. We observe in the experiment that the performance of the algorithm is insensitive to the initial candidate solutions, if the initial variance is large enough. Since CEOCBA uses the elite samples at each iteration to update the sampling distribution, a comparable choice in our algorithms is to set the shape function as $S_{\theta_k}(h(x,\xi)) = I\{h(x,\xi) \geq \gamma_{\theta_k}\}$. In all three algorithms, the quantile parameter ρ is set to be 0.1, and the $(1-\rho)$-quantile γ_{θ_k} is estimated by the $(1-\rho)$ sample quantile of the function values corresponding to all the candidate solutions generated at the k^{th} iteration. In GASSO and GASSO-2T, we set the small constant $\varepsilon = 10^{-10}$, and the step sizes $\alpha_k = 50/(k+2000)^{0.6}$ and $\beta_k = 1/(k+2000)^{0.55}$, which satisfy Assumption 2. The sample size for each iteration is set to be $N = 1000$ for GASSO and $N = 100$ for GASSO-2T, since GASSO-2T can use a small sample size due to the two-timescale updating scheme. For a fair comparison, for CEOCBA we set the total number of sampling budget $T = 1000$, the number of candidate solutions per iteration to be 100, the initial number of function evaluations for each candidate solution $n_0 = 5$, and the budget increment

$\Delta = 100$. For CEOCBA, we apply the smoothing parameter updating procedure in DeBoer et al. (2005) on the parameter updating to prevent premature convergence, and the smoothing parameter is chosen to be $\nu = 0.5$ as suggested by He et al. (2010).

Table 1: Average performance of GASSO, GASSO-2T, and CEOCBA

		GASSO		GASSO-2T		CEOCBA	
	H^*	$\bar{H}^*$	std_err	$\bar{H}^*$	std_err	$\bar{H}^*$	std_err
Powell H_1	-1	-1.025	0.002	-1.195	0.060	-775.9	326.7
Griewank H_2	0	-0.298	0.016	-0.436	0.019	-0.608	0.029
Trigonometric H_3	-1	-1.001	0.522×10^{-4}	-1.003	3.129×10^{-4}	-1.019	0.002
Pinter H_4	-1	-3.010	0.044	-3.809	0.093	-6.486	1.160

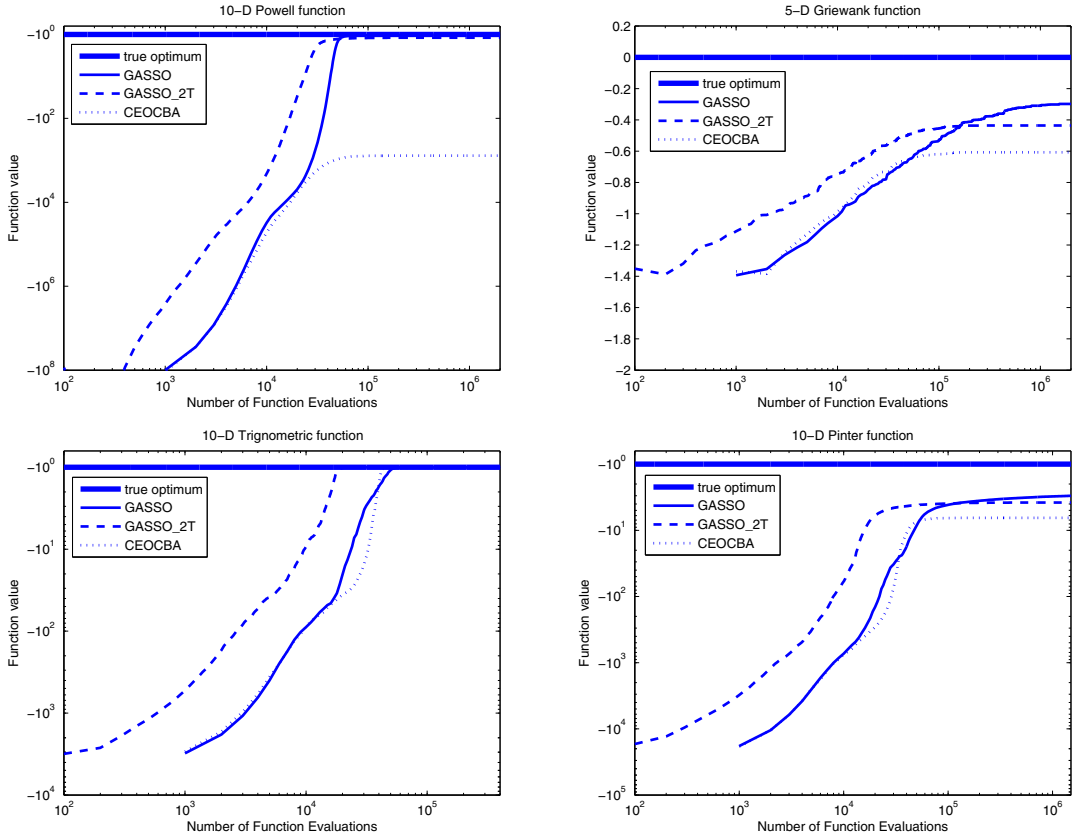

Figure 1: Average performance of GASSO, GASSO-2T, and CEOCBA

We run each of these three algorithms 50 times independently, and compare the average performance in Table 1 and Figure 1. We denote by H^* the true optimal value of $H(\cdot)$, $\bar{H}^*$ the average of the function values $H(\mu_K)$ returned by the 50 runs of an algorithm, where K is the number of iterations of an algorithm, and std_err the standard error of the optimal function values. In the experiments, we found the computation time of function evaluations dominates the time of other steps, so we compare the performance of the algorithms with respect to the total number of function evaluations in Figures 1. For all the test functions, the average optimal function values returned by GASSO and GASSO-2T are very close, although GASSO is slightly more accurate than GASSO-2T. Both return better solutions than CEOCBA on all the test functions. GASSO and CEOCBA have similar convergence speed, but GASSO avoids the problem of allocating simulation budget to candidate solutions and hence is easier to implement. GASSO-2T converges

faster than the other two algorithms and often reduces the total number of function evaluations needed for convergence by about 3-4 times. This confirms the benefit of using a two-timescale updating scheme.

5 CONCLUSION

In this paper, we developed a gradient-based adaptive stochastic search approach to simulation optimization over continuous solution space. Our proposed algorithm iteratively draws candidate solutions from a parameterized sampling distribution and updates the parameter of the sampling distribution using a direct gradient search over the parameter space. Compared with existing model-based methods that often require multiple function evaluations for each candidate solution, a salient feature of our algorithm is that it requires only one single function evaluation per solution, which makes our algorithm simpler to implement without the need to design a simulation budget allocation rule. To reduce the number of candidate solutions that need to be generated per iteration, we further incorporated a two-timescale updating scheme into the algorithm. Numerical results on several benchmark problems showed our algorithms have superior empirical performance, and the two-timescale algorithm improves computational efficiency by several fold.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the National Science Foundation under Grants CMMI-1130273 and Air Force Office of Scientific Research under YIP Grant FA-9550-12-1-0250.

REFERENCES

Andradóttir, S. 2006. "An overview of simulation optimization with random search". In *Handbooks in Operations Research and Management Science: Simulation*, edited by S. Henderson and B. Nelson, Chapter 20, 617–32: Elsevier.

Barton, R., and M. Meckesheimer. 2006. *Handbooks in Operations Research and Management Science: Simulation*, Chapter 18: Metamodel-Based Simulation Optimization, 535 – 574. Elsevier.

Bhatnagar, S., and V. Borkar. 1998. "A Two-Time-Scale Stochastic Approximation Scheme for Simulation Based Parametric Optimization". *Probability in Engineering and Informational Sciences* 12:519 – 531.

Bhatnagar, S., H. L. Prasad, and L. A. Prashanth. 2013. "Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods". In *Lecture Notes in Control and Information Sciences Series*, Volume 434. Springer.

Borkar, V. S. 2008. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Chang, K.-H., L. J. Hong, and H. Wan. 2013. "Stochastic trust-region response-surface method (STRONG) A new response surface framework for simulation optimization". *INFORMS Journal on Computing* 25:230–243.

Chen, C.-H., J. Lin, E. Yucesan, and S. Chick. 2000. "Simulation budget allocation for further enhancing the efficiency of ordinal optimization". *Discrete Event Dynamic Systems: Theory and Applications* 10:251–270.

Chepuri, K., and T. H. de Mello. 2005. "Solving the vehicle routing problem with stochastic demands using the cross entropy method". *Annals of Operations Research* 134:153–181.

de Mello, T. H., A. Shapiro, and M. L. Spearman. 1999. "Finding optimal material release times using simulation-based optimization". *Management Science* 45:86–102.

DeBoer, P. T., D. P. Kroese, S. Mannor, and R. Y. Rubinstein. 2005. "A Tutorial on the Cross-Entropy Method". *Annals of Operations Research* 134:19–67.

Dorigo, M., and L. Gambardella. 1997. "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem". *IEEE Transactions on Evolutionary Computation* 1:53 – 66.

Fu, M., C.-H. Chen, and L. Shi. 2008. "Some Topics for Simulation Optimization". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 27–38. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

He, D., L. H. Lee, C.-H. Chen, M. C. Fu, and S. Wasserkrug. 2010. "Simulation optimization using the cross-entropy method with optimal computing budget allocation". *ACM Transactions on Modeling and Computer Simulation* 20 (1): 4:1–4:22.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. "A Model Reference Adaptive Search Method for Global Optimization". *Operations Research* 55:549–568.

Hu, J., M. C. Fu, and S. I. Marcus. 2008. "A Model Reference Adaptive Search Method for Stochastic Global Optimization". *Commnunications in Information and Systems* 8:245–276.

Kiefer, J., and J. Wolfowitz. 1952. "Stochastic Estimation of the Maximum of a Regression Function". *Annals of Mathematical Statistics* 23:462–466.

Kushner, H. J., and G. G. Yin. 2004. *Stochastic Approximation Algorithms and Applications*. 2nd ed. New York, NY: Springer-Verlag.

Larranaga, P., R. Etxeberria, J. A. Lozano, and J. M. Pena. 1999. "Optimization by learning and simulation of Bayesian and Gaussian networks". Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country.

Molvalioglu, O., Z. B. Zabinsky, and W. Kohn. 2009. "The interacting-particle algorithm with dynamic heating and cooling". *Journal of Global Optimization* 43:329–356.

Olafsson, S. 2006. *Handbooks in operations research and management science*, Chapter 13: Metaheuristics, 633 – 654. Elsevier.

Romeijn, H. E., and R. L. Smith. 1994. "Simulated annealing for constrained global optimization". *Journal of Global Optimization* 5 (2): 101–126.

Wolpert, D. H. 2004. "Finding Bounded Rational Equilibria Part I: Iterative Focusing". In *Proceedings of the Eleventh International Symposium on Dynamic Games and Applications*, edited by T. Vincent.

Zhou, E., and J. Hu. 2012. "Combining Gradient-based Optimization with Stochastic Search". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 208–219. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhou, E., and J. Hu. 2014. "Gradient-based Adaptive Stochastic Search for Non-differentiable Optimization". *IEEE Transactions on Automatic Control* 59 (7): 1818–1832.

AUTHOR BIOGRAPHIES

ENLU ZHOU is an Assistant Professor in the H. Milton Stewart School of Industrial & Systems Engineering at Georgia Institute of Technology. She received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009, and was an assistant professor in the Department of Industrial & Enterprise Systems Engineering at University of Illinois at Urbana-Champaign from 2009 to 2013. Her research interests include stochastic control and simulation optimization, with applications towards Financial engineering. Her email address is enlu.zhou@isye.gatech.edu.

SHALABH BHATNAGAR is Professor in the Computer Science and Automation Department at Indian Institute of Science. His main research interests are in control/optimization in stochastic dynamic systems. Amongst the application domains, he is interested in communication and wireless networks and vehicular traffic control. His email address is shalabh@csa.iisc.ernet.in.

XI CHEN is a Ph.D. candidate in the Department of Industrial & Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. Her research interests lie in the broad areas of simulation optimization. Her email address is xchen37@illinois.edu.