Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

COMBINING GRADIENT-BASED OPTIMIZATION WITH STOCHASTIC SEARCH

Enlu Zhou

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

Jiaqiao Hu

Department of Applied
Mathematics and Statistics
State University of New York
Stony Brook, NY, USA

ABSTRACT

We propose a stochastic search algorithm for solving non-differentiable optimization problems. At each iteration, the algorithm searches the solution space by generating a population of candidate solutions from a parameterized sampling distribution. The basic idea is to convert the original optimization problem into a differentiable problem in terms of the parameters of the sampling distribution, and then use a quasi-Newton-like method on the reformulated problem to find improved sampling distributions. The algorithm combines the strength of stochastic search from considering a population of candidate solutions to explore the solution space with the rapid convergence behavior of gradient methods by exploiting local differentiable structures. We provide numerical examples to illustrate its performance.

1 INTRODUCTION

We consider optimization problems with little structure, and assume that the objective function can only be assessed through "black-box" evaluation, which returns the function value for a specified candidate solution. In such a general setting, there is little problem-specific knowledge that can be exploited in searching for improved solutions. These problems arise in many areas of importance and can be extremely difficult to solve due to the presence of multiple local optimal solutions and the lack of structural properties. An effective and promising approach for tackling such general optimization problems is stochastic search. This refers to a collection of methods that use some sort of randomized mechanism to generate a sequence of iterates, e.g., candidate solutions, and then use the sequence of iterates to successively approximate the optimal solution. Over the past years, various stochastic search algorithms have been proposed in literature. These include approaches such as simulated annealing (Kirkpatrick et al. 1983), genetic algorithms (Goldberg 1989), tabu search (Glover 1990), the nested partitions method (Shi and Ólafsson 2000), pure adaptive search (Zabinsky 2003), sequential Monte Carlo simulated annealing (Zhou and Chen 2012), and the class of model-based algorithms (cf. e.g., Zlochin et al. (2004)).

This paper focuses on model-based algorithms, which construct a sequence of distribution models to characterize promising regions of the solution space. These algorithms typically carry out two interrelated steps at each iteration: (1) draw candidate solutions from the sampling distribution; (2) use the performance 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 the set of optimal solutions. Examples of model-based algorithms include ant colony optimization (Dorigo and Gambardella 1997; Dorigo and Blum 2005), annealing adaptive search (AAS) (Romeijn and Smith 1994), probability collectives (PCs) (Wolpert 2004), the estimation of distribution algorithms (EDAs) (Larranaga et al. 1999; Muhlenbein and Paaß 1996), the cross-entropy (CE) method (Rubinstein 2001), model reference adaptive search (MRAS) (Hu et al. 2007), and the interacting-particle algorithm (Molvalioglu et al. 2009; Molvalioglu et al. 2010). The various model-based algorithms mainly differ in their ways of updating the sampling distribution.

Because model-based algorithms work with a population of candidate solutions at each iteration, they demonstrate more robustness in exploring the solution space as compared with their classical counterparts that work with a single candidate solution each time (e.g., simulated annealing). The main motivation of this paper is to integrate this robustness feature of model-based algorithms into familiar gradient-based tools from classical differentiable optimization to facilitate the search for good sampling distributions. The underlying idea is to reformulate the original (possibly non-differentiable) optimization problem into a *differentiable* optimization problem over the parameter space of the sampling distribution, and then use a direct gradient search method on the parameter space to solve the new formulation. This leads to a natural algorithmic framework that consists of the following two steps at every iteration: (1) generate candidate solutions from the current sampling distribution; (2) update the parameters of the sampling distribution using a direct gradient search method. Although there are a variety of gradient-based algorithms that are applicable in step (2) above, in this paper we focus on a particular algorithm that uses a quasi-Newton-like procedure to update the sampling distribution parameters.

The rest of the paper is organized as follows. We introduce the problem setting formally in Section 2. We propose the algorithm along with its derivations in Section 3. We carry out some numerical study in Section 4 to illustrate the performance of the algorithm. Finally, we conclude this paper in Section 5. All the proofs are given in the Appendix.

2 PROBLEM FORMULATION

Consider the maximization problem

$$x^* \in \arg\max_{x\in\mathscr{X}} H(x), \quad \mathscr{X} \subseteq \mathbb{R}^n. \tag{1}$$

where the solution space $\mathscr{X}$ is a nonempty set in $\mathbb{R}^n$, and $H : \mathscr{X} \to \mathbb{R}$ is a real-valued function. Denote the optimal function value as H^*, i.e., there exists an x^* such that $H(x) \leq H^* \triangleq H(x^*)$, $\forall x \in \mathscr{X}$. Assume that H is bounded on $\mathscr{X}$, i.e., $\exists H_l > -\infty$, $H_u < \infty$ s.t. $H_l \leq H(x) \leq H_u$, $\forall x \in \mathscr{X}$. We consider problems where the objective function $H(x)$ lacks nice structural properties such as differentiability and convexity and could have multiple local optima.

Motivated by the idea of using a sampling distribution in model-based optimization, we let $\{f(x;\theta)|\theta \in \Theta \subseteq \mathbb{R}^d\}$ be a parameterized family of probability density functions (pdfs) on $\mathscr{X}$, where Θ is a parameter space. For each distribution $f(x,\theta)$ in the family, it is easy to see that

$$\int H(x)f(x;\theta)dx \leqslant H^*, \quad \forall\theta \in \mathbb{R}^d.$$

In this paper, we simply write $\int$ with the understanding that the integrals are taken over $\mathscr{X}$. The equality on the right-hand-side is achieved if and only if there exists a θ^* such that the probability mass of $f(x;\theta^*)$ is concentrated only on the set of global optima. Hence, finding x^* can be done through finding θ^*, and the maximization problem (1) can be converted to

$$\theta^* = \arg\max_{\theta\in\mathbb{R}^d} \int H(x)f(x;\theta)dx. \tag{2}$$

So instead of considering directly the original function $H(x)$ that is possibly non-differentiable in x, we now consider the new objective function $\int H(x)f(x;\theta)dx$ that is continuous on the parameter space and usually differentiable with respect to θ. For example, under mild conditions the differentiation can be brought into the integration to apply on the p.d.f. $f(x;\theta)$, which is often differentiable and in particular is differentiable if it is in an exponential family of densities that we will focus on later.

The formulation of (2) suggests a natural integration of stochastic search methods on the solution space $\mathscr{X}$ with gradient-based optimization techniques on the continuous parameter space. Conceptually, that is to iteratively carry out the following two steps:

1. Generate candidate solutions from $f(x;\theta)$ on the solution space $\mathscr{X}$.
2. Use a gradient-based method for the problem (2) to update the parameter θ.

The motivation is to speed up stochastic search with a guidance on the parameter space, and hence combine the advantages of both methods: the fast convergence of gradient-based methods and the global exploration of stochastic search methods. Even though problem (2) may be non-convex and multi-modal in θ, the sampling from the entire original space $\mathscr{X}$ compensates the local exploitation along the gradient on the parameter space. In fact, our algorithm developed later will automatically adjust the magnitude of the gradient step on the parameter space according to the global information, i.e., our belief about the promising regions of the solution space.

For algorithmic development later, 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) $S_\theta(y)$ is upper bounded for bounded y and nondecreasing in y;
(b) The set of optimal solutions $\{\arg\max_{x\in\mathscr{X}} S_\theta(H(x))\}$ is equal to $\{\arg\max_{x\in\mathscr{X}} H(x)\}$, the set of optimal solutions of the original problem (1).

Therefore, solving (1) is equivalent to solving the following problem

$$x^* = \arg\max_{x\in\mathscr{X}} S_\theta(H(x)). \tag{3}$$

The main reason of introducing the shape function S_θ is to ensure nonnegativity of the objective function $S_\theta(H(x))$ under consideration.

For an arbitrary but fixed $\theta' \in \mathbb{R}^d$, define the function

$$L(\theta;\theta') \triangleq \int S_{\theta'}(H(x)) f(x;\theta) dx.$$

According to the conditions on S_θ, it always holds that

$$0 < L(\theta;\theta') \leq S_{\theta'}(H^*) \quad \forall \theta,$$

and the equality is achieved if there exist a θ^* such that the probability mass of $f(x;\theta^*)$ is concentrated only on the set of global optima. Therefore, problem (3) and thus problem (1) can be converted to finding θ^* that solves the following maximization problem:

$$\theta^* = \arg\max_{\theta\in\mathbb{R}^d} L(\theta;\theta'). \tag{4}$$

Same as problem (2), $L(\theta;\theta')$ may be nonconvex and multi-modal in θ.

3 GRADIENT-BASED ADAPTIVE STOCHASTIC SEARCH

Following the formulation in the previous section, 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 quasi-Newton iteration for $\max_\theta L(\theta;\theta_k)$.

3.1 Derivation

We first derive the expressions for the gradient and Hessian of $L(\theta;\theta')$. Assuming it is easy to draw samples from $f(x;\theta)$, then the main obstacle is to find expressions of the gradient and Hessian of $L(\theta;\theta_k)$ that can be nicely estimated using the samples from $f(x;\theta)$. To overcome this obstacle, we choose $\{f(x;\theta)\}$ to be an exponential family of densities defined as below.

Definition 1 Suppose Θ is an open subset in $\mathbb{R}^d$. A family $\{f(x;\theta):\theta\in\Theta\}$ is an exponential family of densities if it satisfies

$$f(x;\theta)=\exp\{\theta^T T(x)-\phi(\theta)\},\ \ \phi(\theta)=\ln\{\int\exp(\theta^T T(x))dx\}. \tag{5}$$

where $T(x)=[T_1(x),T_2(x),\ldots,T_d(x)]^T$ is the vector of sufficient statistics, and $\theta=[\theta_1,\theta_2,\ldots,\theta_d]^T$ is the vector of natural parameters.

Proposition 1 Assume that for any fixed θ', $\theta\mapsto L(\theta;\theta')$ is twice differentiable, and the differentiation is interchangeable with the integration in $L(\theta;\theta')$. If $f(x;\theta)$ is in an exponential family of densities, then

$$\begin{aligned}\nabla_\theta L(\theta;\theta') &= E_\theta[S_{\theta'}(H(X))T(X)]-E_\theta[S_{\theta'}(H(X))]E_\theta[T(X)],\\ \nabla^2_\theta L(\theta;\theta') &= E_\theta\left[S_{\theta'}(H(X))(T(X)-E_\theta[T(X)])(T(X)-E_\theta[T(X)])^T\right]-\mathrm{Var}_\theta[T(X)]E_\theta[S_{\theta'}(H(X))].\end{aligned}$$

Proof. Please see Appendix. □

Notice that if we were to use Newton's method to update the parameter θ, the Hessian $\nabla^2_\theta L(\theta;\theta')$ is not necessarily negative semidefinite to ensure the parameter updating is along the ascent direction of $L(\theta;\theta')$, so we need some stabilization scheme. One way is to approximate the Hessian by the second term on the right-hand-side with a small perturbation, i.e., $-(\mathrm{Var}_\theta[T(X)]+\varepsilon I)E_\theta[S_{\theta'}(H(X))]$, which is always negative definite. Thus, the parameter θ could be updated according to the following iteration

$$\begin{aligned}\theta_{k+1} &= \theta_k+\alpha_k\left((\mathrm{Var}_{\theta_k}[T(X)]+\varepsilon I)E_{\theta_k}[S_{\theta_k}(H(X))]\right)^{-1}\nabla_\theta L(\theta_k;\theta_k),\\ &= \theta_k+\alpha_k\left(\mathrm{Var}_{\theta_k}[T(X)]+\varepsilon I\right)^{-1}\left(\frac{E_{\theta_k}[S_{\theta_k}(H(X))T(X)]}{E_{\theta_k}[S_{\theta_k}(H(X))]}-E_{\theta_k}[T(X)]\right),\end{aligned} \tag{6}$$

where $\alpha_k>0$ is the step size, and E_{θ_k} and Var_{θ_k} denote the expectation and variance taken with respect to $f(\cdot;\theta_k)$, respectively. Define a density function

$$p(x;\theta)\triangleq\frac{S_\theta(H(x))f(x;\theta)}{\int S_\theta(H(x))f(x;\theta)dx}=\frac{S_\theta(H(x))f(x;\theta)}{L(\theta;\theta)}. \tag{7}$$

Then (6) can be rewritten as

$$\theta_{k+1}=\theta_k+\alpha_k(\mathrm{Var}_{\theta_k}[T(X)]+\varepsilon I)^{-1}\left(E_{p_k}[T(X)]-E_{\theta_k}[T(X)]\right), \tag{8}$$

where E_{p_k} denotes the expectation with respect to $p(\cdot;\theta_k)$.

In the updating equation (8), the term $E_{\theta_k}[S_{\theta_k}(H(X))]^{-1}$ is absorbed into $\nabla_\theta L(\theta_k;\theta_k)$, so we obtain a scale-free term $(E_{p_k}[T(X)]-E_{\theta_k}[T(X)])$ that is not subject to the scaling of the function value of $S_{\theta_k}(H(x))$. It would be nice to have such a scale-free gradient so that we can employ other gradient-based methods more easily besides the above specific choice of a quasi-Newton method. Towards this direction, we consider a further transformation of the maximization problem (4) by letting

$$l(\theta;\theta')=\ln L(\theta;\theta').$$

Since $\ln:R^+\to R$ is a strictly increasing function, the maximization problem (4) is equivalent to

$$\theta^*=\arg\max_{\theta\in\mathbb{R}^d} l(\theta;\theta'). \tag{9}$$

The gradient and the Hessian of $l(\theta;\theta')$ are given in the following proposition.

Proposition 2 Assume that for any fixed θ', $\theta \mapsto l(\theta;\theta')$ is twice differentiable, and the differentiation is interchangeable with the integration in $l(\theta;\theta')$. If $f(x;\theta)$ is in an exponential family of densities, then

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

where $p(\cdot;\theta')$ is as defined in (7).

Proof. Please see Appendix. □

Similarly as before, 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., $-(\mathrm{Var}_{\theta'}[T(X)]+\varepsilon I)$. Then by setting

$$\theta_{k+1} = \theta_k + \alpha_k \left(\mathrm{Var}_{\theta_k}[T(X)] + \varepsilon I\right)^{-1} \nabla_\theta l(\theta_k),$$

we again obtain exactly the same updating equation (8) for θ. The difference from (6) is that the gradient $\nabla_\theta l(\theta;\theta')$ is a scale-free term, and hence can be used in other gradient-based methods with easier choices of the step size. From an algorithmic viewpoint, it is better to consider the optimization problem (9) on $l(\theta;\theta')$ instead of the problem (4) on $L(\theta;\theta')$, even though both have the same global optima.

Although there are many ways to determine the positive definite matrix in front of the gradient in a quasi-Newton method, our choice of $(\mathrm{Var}_{\theta_k}[T(X)]+\varepsilon I)^{-1}$ is not arbitrary but based on some principle. Without considering the numerical stability and thus dropping the term εI, the term $\mathrm{Var}_\theta[T(X)] = E[\nabla_\theta \ln f(X;\theta)(\nabla_\theta \ln f(X;\theta))^T] = E[-\nabla^2_\theta \ln f(X;\theta)]$ is the Fisher information matrix, whose inverse provides a lower bound on the covariance matrix of an unbiased estimator of the parameter θ (in the sense that the latter matrix subtracting the former one is a positive semi-definite matrix) (Rao 1945), leading to the fact that $(\mathrm{Var}_\theta[T(X)])^{-1}$ is the minimum-variance step size in stochastic approximation. Moreover, from the optimization perspective, the term $(\mathrm{Var}_\theta[T(X)])^{-1}$ relates the gradient search on the parameter space with the stochastic search on the solution space, and thus adaptively adjusts the updating of the sampling distribution to our belief about the promising regions of the solution space. To see this more easily, let us consider the special case $T(x)=x$. Then, a small value of $(\mathrm{Var}_\theta[X])^{-1}$ indicates that the sampling distribution $f(\cdot;\theta)$ has a large variance (i.e., exploration dominates exploitation), so the algorithm is more conservative by allowing a small amount of update in θ_k. On the other hand, a large value of $(\mathrm{Var}_\theta[X])^{-1}$ indicates that we are more confident about certain promising regions, so a large gradient step in updating θ is taken.

3.2 Algorithm

We will use the Quasi-Newton scheme (8) to update the parameter θ of the sampling distribution. In implementation, the term $E_{p_k}[T(X)]$ is often not analytically available and needs to be estimated. Suppose $\{x_k^1,\ldots,x_k^{N_k}\}$ are independent and identically distributed (i.i.d.) samples drawn from $f(x;\theta_k)$. Since

$$E_{p_k}[T(X)] = E_{\theta_k}\left[T(X)\frac{p(X;\theta_k)}{f(X;\theta_k)}\right],$$

we compute the weights $\{w_k^i\}$ for the samples $\{x_k^i\}$ according to

$$\begin{aligned}& w_k^i \propto \frac{p(x_k^i;\theta_k)}{f(x_k^i;\theta_k)} \propto S_{\theta_k}(H(x_k^i)), \quad i=1,\ldots,N_k,\\ & \sum_{i=1}^{N} w_k^i = 1.\end{aligned}$$

Hence, $E_{p_k}[T(X)]$ can be approximated by

$$\widetilde{E}_{p_k}[T(X)] = \sum_{i=1}^{N_k} w_k^i T(x_k^i). \tag{10}$$

Some forms of the function $S_{\theta_k}(H(x))$ have to be approximated by samples as well. For example, if $S_{\theta_k}(H(x))$ takes the form $S_{\theta_k}(H(x)) = (H(x) - H_l)I\{H(x) \geq \gamma_{\theta_k}\}$, where γ_θ is the $(1-\rho)$-quantile, then the quantile γ_{θ_k} needs to be estimated by the sample quantile. In this case, we denote the approximation by $\widehat{S}_{\theta_k}(H(x))$, and evaluate the normalized weights according to

$$\widehat{w}_k^i = \frac{\widehat{S}_{\theta_k}(H(x_k^i))}{\sum_{j=1}^{N_k} \widehat{S}_{\theta_k}(H(x_k^j))}, \quad i = 1, \ldots, N_k.$$

Then the term $E_{p_k}[T(X)]$ is approximated by

$$\widehat{E}_{p_k}[T(X)] = \sum_{i=1}^{N_k} \widehat{w}_k^i T(x_k^i). \tag{11}$$

The variance term $\mathrm{Var}_{\theta_k}[T(X)]$ is either not directly available or too complicated to compute analytically, so it also needs to be estimated by samples:

$$\widehat{\mathrm{Var}}_{\theta_k}[T(X)] \quad = \quad \frac{1}{N_k - 1}\sum_{i=1}^{N_k} T(x_k^i)T(x_k^i)^T - \frac{1}{N_k^2 - N_k}\left(\sum_{i=1}^{N_k} T(x_k^i)\right)\left(\sum_{i=1}^{N_k} T(x_k^i)\right)^T. \tag{12}$$

The expectation term $E_{\theta_k}[T(X)]$ can be evaluated analytically in most cases. For example, when $\{f(\cdot;\theta_k)\}$ is chosen as the Gaussian family, $E_{\theta_k}[T(X)]$ reduces to the mean and second moment of the Gaussian distribution.

Based on the above implementation of the updating scheme (8) for θ, we propose the following algorithm, namely Gradient-based Adaptive Stochastic Search (GASS), for solving the maximization problem (1).

Algorithm 1 Gradient-Based Adaptive Stochastic Search (GASS)

1. *Initialization:* choose an exponential family of densities $\{f(\cdot;\theta)\}$, and specify a small positive constant ε, initial parameter θ_0, sample size $\{N_k\}$, and step size $\{\alpha_k\}$. Set $k = 0$.
2. *Sampling:* draw samples $x_k^i \overset{\text{iid}}{\sim} f(x;\theta_k), i = 1, 2, \ldots, N_k$.
3. *Estimation:* compute the normalized weights $\widehat{w}_k^i$ according to

$$\widehat{w}_k^i = \frac{\widehat{S}_{\theta_k}(H(x_k^i))}{\sum_{j=1}^{N_k} \widehat{S}_{\theta_k}(H(x_k^j))},$$

   and then compute $\widehat{E}_{p_k}[T(X)]$ and $\widehat{\mathrm{Var}}_{\theta_k}[T(X)]$ respectively according to (11) and (12).
4. *Updating:* update the parameter θ according to

$$\theta_{k+1} = \theta_k + \alpha_k(\widehat{\mathrm{Var}}_{\theta_k}[T(X)] + \varepsilon I)^{-1}(\widehat{E}_{p_k}[T(X)] - E_{\theta_k}[T(X)]).$$

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 above algorithm, at the k^{th} iteration candidate solutions are drawn from the sampling distribution $f(\cdot;\theta_k)$, and then are used to estimate the quantities in the updating equation for θ so as to generate the next sampling distribution $f(\cdot;\theta_{k+1})$. For the ease of exposition, suppose $T(X)=X$, and then the term $\widehat{\mathrm{Var}}_{\theta_k}[T(X)]$ basically measures how widespread the candidate solutions are. Since the magnitude of the ascent step is determined by $(\widehat{\mathrm{Var}}_{\theta_k}[T(X)]+\varepsilon I)^{-1}$, the algorithm takes smaller ascent steps to update θ while the candidate solutions are more widely spread (i.e., $\widehat{\mathrm{Var}}_{\theta_k}[X]$ is larger), and takes larger ascent steps while the candidate solutions are more concentrated (i.e., $\widehat{\mathrm{Var}}_{\theta_k}[X]$ is smaller). It means that exploitation of the local structure is adapted to our belief about the promising regions of the solution space: we will be more conservative in exploitation if we are uncertain about where the promising regions are and more bold otherwise.

3.3 Accelerated GASS

GASS can be viewed as a stochastic approximation (SA) algorithm in searching the root of

$$(\mathrm{Var}_{\theta_k}[T(X)]+\varepsilon I)^{-1}\nabla_\theta l(\theta;\theta_k)|_{\theta=\theta_k}=0.$$

To improve the convergence rate of SA algorithms, Polyak (1990) and Ruppert (1991) first proposed to take the average of the θ values generated by previous iterations, which is often referred to as Polyak (or Polyak-Ruppert) averaging. The original Polyak averaging technique is "offline", i.e., the averages are not fed back into the iterates of θ, and hence the averages are not useful for guiding the stochastic search in our context. However, there is a variation, Polyak averaging with online feedback (c.f. pp. 75 - 76 in Kushner and Yin (2004)), which could also enhance the convergence rate of SA, although not as optimal as the original Polyak averaging. It can be applied to accelerate the convergence of GASS, i.e., the parameter θ will be updated according to

$$\theta_{k+1}=\theta_k+\alpha_k\left(\widehat{\mathrm{Var}}_{\theta_k}[T(X)]+\varepsilon I\right)^{-1}(\widehat{E}_{p_k}[T(X)]-E_{\theta_k}[T(X)])+\alpha_k c(\bar{\theta}_k-\theta_k), \tag{13}$$

where the constant c is the feedback weight, and $\bar{\theta}_k$ is the average

$$\bar{\theta}_k=\frac{1}{k}\sum_{i=1}^{k}\theta_i,$$

which can be calculated recursively by

$$\bar{\theta}_k=\frac{k-1}{k}\bar{\theta}_{k-1}+\frac{\theta_k}{k}. \tag{14}$$

With this parameter updating scheme, we propose the accelerated GASS algorithm as following.

Algorithm 2 Gradient-based Adaptive Stochastic Search with Averaging (GASS_avg)
Same as Algorithm 1 except in step 4) the parameter updating follows (13) and (14).

4 NUMERICAL EXPERIMENTS

In this section, we test the proposed algorithms GASS and GASS_avg on some benchmark continuous optimization problems selected from (Hu et al. 2007). To fit in the maximization framework where our algorithms are proposed, we consider the negative of those objective functions that are originally for minimization problems. The problems we consider are listed below with their dimensions in the parentheses.

(1) Griewank function (n=20): $H_1(x)=-\frac{1}{4000}\sum_{i=1}^{n}x_i^2+\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)-1$, where $x^*=(0,\cdots,0)^T$, $H^*=0$.

(2) Trigonometric function (n=20): $H_2(x) = -\sum_{i=1}^{n}\left[8\sin^2(7(x_i-0.9)^2)+6\sin^2(14(x_i-0.9)^2)+(x_i-0.9)^2\right]-1$, where $x^* = (0.9,\cdots,0.9)^T$, $H^* = -1$.

(3) Powel singular function (n=20): $H_3(x) = -\sum_{i=2}^{n-2}\left[(x_{i-1}+10x_i)^2+5(x_{i+1}-x_{i+2})^2+(x_i-2x_{i+1})^4+10(x_{i-1}-x_{i+2})^4\right]-1$, where $x^* = (0,\cdots,0)^T$, $H^* = -1$.

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

Specifically, Griewank (H_1) and Trigonometric (H_2) are multimodal problems with a large number of local optima, and the number of local optima increases exponentially with the problem dimension; Powel (H_3) is a badly-scaled function; Pintér (H_4) is both multimodal and badly-scaled.

We compare the performance of GASS and GASS_avg with two other algorithms: the latest version of the cross-entropy (CE) method, i.e., the modified CE based on stochastic approximation (Hu et al. 2012), and Model Reference Adaptive Search (MRAS) (Hu et al. 2007). We choose the shape function in GASS and GASS_avg to be of the similar form as in the CE method or MRAS:

$$S_{\theta_k}(H(x)) = (H(x)-H_l)I\{H(x) \geq \gamma_{\theta_k}\},$$

where 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 all the four methods, we set $\rho = 0.05$, set the sample size $N = 1000$, and choose the parameterized exponential family of distributions $f(x;\theta_k)$ to be multivariate normal distributions $\mathcal{N}(\mu_k,\Sigma_k)$, with the initial mean μ_0 generated randomly according to the uniform distribution on $[-50,50]^n$, and the initial covariance matrix set to be $\Sigma_0 = 2500I_{n\times n}$, where n is the dimension of the problem. We observe in numerical experiments that the performance of the algorithms is insensitive to the choice of the initial candidate solutions if the initial variance is large. In GASS and GASS_avg, we use the step size $\alpha_k = \frac{\alpha_0}{(k+A)^\alpha}$, where α_0 reflects the initial step size, the constant A is used to keep the initial step size small and reduce the unstable behavior, and the parameter α should be in $(0,1]$; we set $A = 50$, $\alpha_0 = 10$, and $\alpha = 0.5$ for all the problems. In GASS_avg, the feedback weight is $c = 0.1$ for problems H_1 and H_2 and $c = 0.02$ for H_3 and H_4. In the modified CE method, we use the gain sequence $\alpha_k = 5/(k+100)^{0.5}$, which should be less than 1 for all k. In the implementation of MRAS, we use a smoothing parameter ν when updating the parameter θ_k of the exponential family of distributions; we set $\nu = 0.2$ as suggested by Hu et al. (2007); the rest of the parameters in MRAS are set as follows: $\varepsilon = 10^{-5}$, $\lambda = 0.01$, and $r = 10^{-4}$.

Table 1: Comparison of GASS, GASS_avg, Modified CE and MRAS.

		GASS		GASS_avg		Modified CE		MRAS	
	H^*	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε	$\bar{H}^*(std\_err)$	M_ε
Griewank H_1	0	0(3.75E-14)	100	0(4.48E-15)	100	0(4.22E-10)	100	-0.034(0.0044)	7
Trigonometric H_2	-1	-1(4.67E-13)	100	-1(6.21E-13)	100	-1(3.03E-12)	100	-1.063(0.0216)	90
Powel H_3	-1	-1(3.19E-7)	100	-1(9.9E-7)	100	-1(1.18E-15)	100	-1.0002(4.66E-6)	100
Pinter H_4	-1	-1.001(1.47E-5)	100	-1.013(0.006)	91	-1(2.23E-18)	100	-1.101(0.0076)	2

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, which is equal to the total number of samples. The average performance based on 100 independent runs for each method is shown in Table 1, where H^* is the true optimal function value; $\bar{H}^*$ is the average of the function values returned by the 100 runs of an algorithm; *std_err* is the standard error of these 100 function values; M_ε is the number of ε-optimal solutions out of 100 runs (ε-optimal solution is the solution such that $H^* - \hat{H}^* \leq \varepsilon$, where $\hat{H}^*$ is the optimal function value returned by an algorithm). We consider $\varepsilon = 10^{-2}$ for problem H_4 and $\varepsilon = 10^{-3}$ for all other problems. Fig. 1 shows the average (over the 100 runs) best value of $H(\cdot)$ at current iteration versus the total number of samples generated so far.

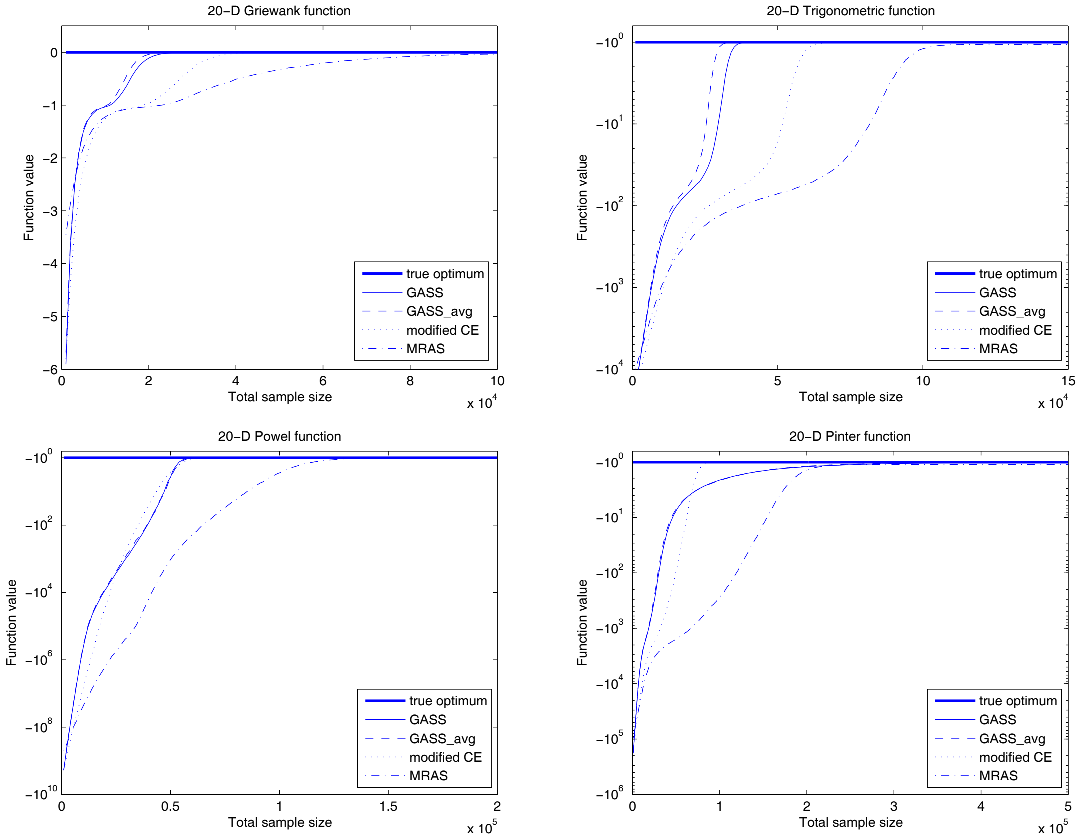

Figure 1: Comparison of GASS, GASS_avg, Modified CE and MRAS.

From the results, GASS and modified CE find the ε-optimal solutions in all the 100 runs for all the problems; GASS_avg finds all the ε-optimal solutions for H_1, H_2, and H_3; MRAS only finds all the ε-optimal solutions for the badly-scaled problem H_3. GASS_avg always converges faster than GASS, verifying the effectiveness of the averaging of iterates with online feedback. Both GASS and GASS_avg converge faster than MRAS on all the problems, and converge faster than the modified CE method on most problems except H_4.

5 CONCLUSION

In this paper, we introduced a new algorithm, Gradient-based Adaptive Stochastic Search (GASS), for solving general optimization problems with little structure. The algorithm generates candidate solutions from a parameterized sampling distribution over the feasible region, and uses a quasi-Newton-like iteration on the parameter space of the parameterized distribution to find improved sampling distributions. Thus, the algorithm enjoys the fast convergence speed of classical gradient search methods while retaining the robustness feature of model-based optimization methods. We further developed an accelerated version of GASS (GASS_avg), relying on the Polyak avaraging technique with online feedback. Our preliminary numerical experiments show that both GASS and GASS_avg work very well on several benchmark problems.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grants ECCS-0901543, CMMI-1130273, CMMI-0900332, CMMI-1130761, and by the Air Force Office of Scientific Research under

Grants FA95501010340 and FA9550-12-1-0250. We are grateful to Xi Chen, a graduate student in the Department of Industrial & Enterprise Systems Engineering at UIUC, for her help with conducting the numerical experiments in Section 4.

A APPENDIX

Proof of Proposition 1. Consider the gradient of $L(\theta;\theta')$ with respect to θ,

$$\begin{aligned}
\nabla_\theta L(\theta;\theta') &= \int S_{\theta'}(H(x))\nabla_\theta f(x;\theta)dx \\
&= \int S_{\theta'}(H(x))f(x;\theta)\nabla_\theta \ln f(x;\theta)dx \\
&= E_\theta[S_{\theta'}(H(X))\nabla_\theta \ln f(X;\theta)]. \qquad (15)
\end{aligned}$$

Consider the Hessian of $L(\theta;\theta')$ with respect to θ,

$$\begin{aligned}
\nabla^2_\theta L(\theta;\theta') &= \int S_{\theta'}(H(x))\nabla^2_\theta f(x;\theta)dx \\
&= \int S_{\theta'}(H(x))f(x;\theta)\nabla^2_\theta \ln f(x;\theta)dx + \int S_{\theta'}(H(x))\nabla_\theta \ln f(x;\theta)\nabla_\theta f(x;\theta)^T dx \\
&= E_\theta[S_{\theta'}(H(X))\nabla^2_\theta \ln f(X;\theta)] + E_\theta[S_{\theta'}(H(X))\nabla_\theta \ln f(x;\theta)\nabla_\theta \ln f(x;\theta)^T], \qquad (16)
\end{aligned}$$

where the last equality follows from the fact that $\nabla_\theta f(x;\theta) = f(x;\theta)\nabla_\theta \ln f(x;\theta)$. Furthermore, if $f(x;\theta) = \exp\{\theta^T T(x) - \phi(\theta)\}$, we have

$$\begin{aligned}
\nabla_\theta \ln f(x;\theta) &= \nabla_\theta \left(\theta^T T(x) - \ln \int \exp(\theta^T T(x))dx\right) \\
&= T(x) - \frac{\int \exp(\theta^T T(x))T(x)dx}{\int \exp(\theta^T T(x))dx} \\
&= T(x) - E_\theta[T(X)]. \qquad (17)
\end{aligned}$$

Plugging (17) into (15) yields

$$\nabla_\theta L(\theta;\theta') = E_\theta[S_{\theta'}(H(X))T(X)] - E_\theta[S_{\theta'}(H(X))]E_\theta[T(X)].$$

Differentiating (17) with respect to θ, we obtain

$$\begin{aligned}
\nabla^2_\theta \ln f(x;\theta) &= -\frac{\int \exp(\theta^T T(x))T(x)T(x)^T dx}{\int \exp(\theta^T T(x))dx} \\
&\quad + \frac{\int \exp(\theta^T T(x))T(x)dx \left(\int \exp(\theta^T T(x))T(x)dx\right)^T}{\left(\int \exp(\theta^T T(x))dx\right)^2} \\
&= -E_\theta[T(X)T(X)^T] + E_\theta[T(X)]E_\theta[T(X)]^T \\
&= -\mathrm{Var}_\theta[T(X)]. \qquad (18)
\end{aligned}$$

Plugging (17) and (18) into (16) yields

$$\nabla^2_\theta L(\theta;\theta') = E_\theta[S_{\theta'}(H(X))(T(X) - E_\theta[T(X)])(T(X) - E_\theta[T(X)])^T] - \mathrm{Var}_\theta[T(X)]E_\theta[S_{\theta'}(H(X))].$$

Proof of Proposition 2. Consider the gradient of $l(\theta;\theta')$ with respect to θ,

$$\begin{aligned}\nabla_\theta l(\theta;\theta')|_{\theta=\theta'} &= \left.\frac{\nabla_\theta L(\theta;\theta')}{L(\theta;\theta')}\right|_{\theta=\theta'}\\ &= \left.\frac{\int S_{\theta'}(H(x))f(x;\theta)\nabla_\theta \ln f(x;\theta)dx}{L(\theta;\theta')}\right|_{\theta=\theta'} \\ &= E_{p(\cdot;\theta')}[\nabla_\theta \ln f(X;\theta')].\end{aligned} \tag{19}$$

Differentiating (19) with respect to θ, we obtain the Hessian

$$\begin{aligned}\nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} &= \frac{\int S_{\theta'}(H(x))f(x;\theta)\nabla^2_\theta \ln f(x;\theta)dx}{L(\theta;\theta')}+\frac{\int S_{\theta'}(H(x))\nabla_\theta \ln f(x;\theta)(\nabla_\theta f(x;\theta))^T dx}{L(\theta;\theta')}\cdots\\ &- \left.\frac{(\int S_{\theta'}(H(x))f(x;\theta)\nabla_\theta \ln f(x;\theta)dx)(\nabla_\theta L(\theta;\theta'))^T}{L(\theta;\theta')^2}\right|_{\theta=\theta'}\end{aligned}$$

Using $\nabla_\theta f(x;\theta)=f(x;\theta)\nabla_\theta \ln f(x;\theta)$ in the second term on the right-hand-side, the above expression can be written as

$$\begin{aligned}\nabla^2_\theta l(\theta;\theta')|_{\theta=\theta'} &= E_{p(\cdot;\theta')}[\nabla^2_\theta \ln f(X;\theta')]+E_{p(\cdot;\theta')}\left[\nabla_{\theta'} \ln f(X;\theta')(\nabla_{\theta'} \ln f(X;\theta'))^T\right]\\ &- E_{p(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right]E_{p(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right]^T\\ &= E_{p(\cdot;\theta')}[\nabla^2_\theta \ln f(X;\theta')]+\mathrm{Var}_{p(\cdot;\theta')}\left[\nabla_\theta \ln f(X;\theta')\right].\end{aligned} \tag{20}$$

Furthermore, if $f(x;\theta)=\exp\{\theta^T T(x)-\phi(\theta)\}$, plugging (17) into (19) yields

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

and plugging (17) and (18) into (20) yields

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

REFERENCES

Dorigo, M., and C. Blum. 2005. “Ant Colony Optimization Theory: a Survey”. *Theoretical Computer Science* 344:243 – 278.

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.

Glover, F. W. 1990. “Tabu Search: A Tutorial”. *Interfaces* 20:74 – 94.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

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., P. Hu, and H. Chang. 2012. “A Stochastic Approximation Framework for a Class of Randomized Optimization Algorithms”. *IEEE Transactions on Automatic Control* 57 (1): 165 – 178.

Kirkpatrick, S., C. D. Gelatt, and J. M. P. Vecchi. 1983. “Optimization by Simulated Annealing”. *Science* 220:671–680.

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.

Molvalioglu, O., Z. B. Zabinsky, and W. Kohn. 2010. “Meta-Control of an Interacting-Paricle Algorithm”. *Nonlinear Analysis: Hybrid Systems* 4 (4): 659 – 671.

Muhlenbein, H., and G. Paaß. 1996. “From Recombination of Genes to the Estimation of Distributions: I. Binary Parameters”. In *Parallel Problem Solving from Nature-PPSN IV*, edited by H. M. Voigt, W. Ebeling, I. Rechenberg, and H. P. Schwefel, 178–187. Berlin, Germany: Springer Verlag.

Polyak, B. 1990. “New Stochastic Approximation Type Procedures”. *Automation and Remote Control* 51:937–946.

Rao, C. 1945. “Information and Accuracy Attainable in the Estimation of Statistical Parameters”. *Bulletin of the Calcutta Mathematical Society* 37:81–91.

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

Rubinstein, R. Y. 2001. “Combinatorial Optimization, Ants and Rare Events”. In *Stochastic Optimization: Algorithms and Applications*, edited by S. Uryasev and P. Pardalos, 304–358. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Ruppert, D. 1991. “Stochastic Approximation”. In *Handbook in Sequential Analysis*, edited by B. G. . P. Sen, 503 – 529.

Shi, L., and S. Ólafsson. 2000. “Nested Partitions Method for Global Optimization”. *Operations Research* 48 (3): 390 – 407.

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.

Zabinsky, Z. B. 2003. *Stochastic Adaptive Search for Global Optimization*. Nonconvex Optimization and Its Applications. Springer.

Zhou, E., and X. Chen. 2012. “Sequential Monte Carlo Simulated Annealing”. *Journal of Global Optimization*. Forthcoming. DOI: 10.1007/s10898-011-9838-3.

Zlochin, M., M. Birattari, N. Meuleau, and M. Dorigo. 2004. “Model-based search for combinatorial optimization: A critical survey”. *Annals of Operations Research* 131:373–395.

AUTHOR BIOGRAPHIES

ENLU ZHOU is an Assistant Professor in the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. 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. Her research interests include stochastic control and simulation optimization, with applications towards financial engineering. Her email address is enluzhou@illinois.edu.

JIAQIAO HU is an Associate Professor in the Department of Applied Mathematics and Statistics at the State University of New York, Stony Brook. He received the B.S. degree in automation from Shanghai Jiao Tong University, the M.S. degree in applied mathematics from the University of Maryland, Baltimore County, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park. His research interests include Markov decision processes, applied probability, and simulation optimization. His email address is jqhu@ams.sunysb.edu.