Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

OPTIMIZING CONDITIONAL VALUE-AT-RISK VIA GRADIENT-BASED ADAPTIVE STOCHASTIC SEARCH

Helin Zhu
Joshua Hale
Enlu Zhou

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
755 Ferst Drive, NW,
Atlanta, GA 30332, USA

ABSTRACT

Optimizing risk measures such as Conditional Value-at-Risk (CVaR) is often a difficult problem, because 1) the loss function might lack structural properties such as convexity or differentiability, since it is usually generated via black-box simulation of a stochastic system; 2) evaluation of CVaR usually requires rare-event simulation, which is computationally expensive. In this paper, we study the extension of the recently proposed gradient-based adaptive stochastic search (GASS) method to the optimization of CVaR. Instead of optimizing CVaR at the risk level of interest directly, we propose to initialize the algorithm at a small risk level, and then increase the risk level at each iteration adaptively until the target risk level is achieved, while the algorithm converges to an optimal solution of the original problem. It enables us to adaptively reduce the number of samples needed to estimate the CVaR at each iteration, and improves the overall efficiency of the algorithm.

1 INTRODUCTION

Risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) are widely studied in various fields to quantify the extreme behaviors of functions of interest. Loosely speaking, VaR characterizes the tail (e.g., 99%) quantile of a distribution, and CVaR characterizes the conditional mean of the tail portion of a distribution. VaR, as one of the earliest risk measures introduced in financial risk management, is easy to understand and interpret for practitioners. CVaR, as a classic coherent risk measure (see, e.g., Artzner et al. (1999)), exhibits nice properties such as convexity and monotonicity for optimization. They have been extensively used in the financial industry, especially after the financial crisis in 2008. An abundant literature has dedicated to studying the estimation and optimization of risk measures under various settings, see, e.g., Rockafellar and Uryasev (2000), Rockafellar and Uryasev (2002), Alexander et al. (2006), Trindade et al. (2007), Hong (2009), Hong and Liu (2009), etc.

In general, optimizing risk measures over continuous decision variables is a challenging problem, especially when the underlying loss function does not possess good structural properties such as convexity or differentiability. Traditional gradient-based optimization methods often are not applicable, since little problem-specific knowledge is known when the loss function is evaluated via black-box simulation of a stochastic system. In contrast, model-based optimization methods are good alternatives as they impose minimal assumptions on the problem structure, which include annealing adaptive search (AAS) (Romeijn and Smith (1994)), the cross-entropy (CE) method (Rubinstein (2001)), model reference adaptive search (MRAS) (Hu et al. (2007)), gradient-based adaptive stochastic search (GASS) (Zhou and Hu (2014)), etc.

The main idea of model-based methods is to introduce a sampling distribution, which often belongs to a parameterized family of densities, over the solution space, and iteratively update the sampling distribution

(or its parameter) by drawing and evaluating candidate solutions according to the sampling distribution. The hope is to have the sampling distribution more and more concentrated on the promising region of the solution space where the optimal solutions are located, and eventually become a degenerate distribution on one of the global optima. Therefore, finding an optimal solution in the solution space is transformed to finding an optimal sampling distribution parameter in the parameter space. A key difference among the aforementioned model-based methods lies in how to update the sampling distribution. For example, in MRAS and GASS, the updating rule on the sampling distribution parameter is derived by converting the original (possibly non-differentiable) deterministic optimization problem into a differentiable stochastic optimization problem on the sampling distribution parameter, and then applying stochastic approximation schemes. Compared with gradient-based methods, model-based methods are more robust in the sense that at every iteration they exploit the promising region of the solution space that has already been identified, while maintaining the exploration of the entire solution space. The updating rule on the sampling distribution parameter controls the balance between the exploration and the exploitation.

Although all the aforementioned model-based methods are designed for deterministic optimization problems, they can be extended to risk (CVaR or VaR) optimization problems in which the exact risk values are replaced with sample estimates (though might be biased). However, a straightforward extension usually leads to an algorithm that is computationally expensive, due to the rare-event simulation required in estimating the risk values. This issue is even more severe when a large risk level (close to 1) is of interest. It inspires us to think about the following question: is it possible to initialize a model-based algorithm for a risk optimization problem with a small risk level (close to 0), and then adaptively adjust or increase the risk level at every iteration until the target risk level is achieved, while the algorithm converges to an optimal solution of the original problem? The hope is that the algorithm will consume less simulation budget (since the risk level is small) during the "warm-up" phase of the algorithm, solve problems that are close to the original one during the "convergence" phase of the algorithm, and eventually achieve budget saving. The key to this question lies in finding a way to link the updating rule on the risk level with the updating rule on the sampling distribution parameter.

In this paper, we will focus on the extension of a specific model-based methods—GASS by Zhou and Hu (2014) to the optimization of risk measures. We choose GASS because it could also be interpreted as a gradient-based scheme of a reformulated problem, in which a Newton-like updating rule is applied on the sampling distribution parameter, and thus the gradient (even the Hessian) of the reformulated problem can be viewed as a signal that empirically measures the algorithm's emphasis between the exploration of the entire solution space and the exploitation of the promising region. Therefore, we could adjust the risk level adaptively using the information contained in the gradient (e.g., its norm) at every iteration. In particular, we will propose an updating rule that increases the risk level proportionally to the decrease in the norm of the gradient. To the best of our knowledge, this work is among the first to apply model-based algorithms to risk optimization problems, and among the first to propose a risk optimization scheme that adaptively adjusts the risk level. Due to the space limit, we will only focus on CVaR optimization, and the extension of the proposed algorithm to VaR optimization (and possibly other risk measures such as probability of large loss) is straightforward. We also leave the convergence analysis of the proposed algorithm to future work.

The rest of the paper is organized as follows. In Section 2, we will formulate the CVaR optimization problem. Then we extend GASS algorithm, which is originally developed for deterministic non-differentiable optimization problems, to the CVaR optimization problem. The detailed algorithms are presented in Section 3, in which Algorithm 1 (referred to as "GASS-CVaR") is a straightforward extension of GASS and Algorithm 2 (referred to as "GASS-CVaR-ARL") further incorporates an updating rule for adaptive risk level adjustments. In Section 4, we illustrate the performance of the proposed algorithms by carrying out numerical tests on several benchmark loss functions. We conclude the paper in Section 5.

2 GENERAL FRAMEWORK

Consider a scalar loss function of the form $l(x,\xi_x)$, where $x \in \mathscr{X} \subseteq \mathbb{R}^{d_x}$ represents the decision variables, and ξ_x represents the randomness in the loss function and its distribution may or may not depend on x. The loss function $l(x,\xi_x)$ can be evaluated either directly or through simulation. Furthermore, to ease the presentation, assume $l(x,\xi_x)$ admits an almost everywhere (a.e.) positive and continuous probability density function (p.d.f.) $p(t;x)$, and thus a continuous and strictly increasing cumulative distribution function (c.d.f.) $P(t;x)$ for all $x \in \mathscr{X}$. The objective is to minimize the CVaR of the loss function $l(x;\xi_x)$ at a risk level of interest α^* $(0 < \alpha^* < 1)$ with respect to (w.r.t.) $x \in \mathscr{X}$. That is, to solve the following stochastic optimization problem:

$$\min_{x\in\mathscr{X}} C_{\alpha^*,l}(x) \stackrel{\triangle}{=} CVaR_{\alpha^*}\left(l(x,\xi_x)\right), \quad \text{or equivalently,} \quad \max_{x\in\mathscr{X}} -C_{\alpha^*,l}(x), \tag{1}$$

where $CVaR_{\alpha^*}\left(l(x,\xi_x)\right)$ is defined by

$$\begin{aligned} CVaR_{\alpha^*}\left(l(x,\xi_x)\right) &\stackrel{\triangle}{=} \mathbb{E}_{\xi_x}\left[l(x,\xi_x)|l(x,\xi_x) \geq V_{\alpha^*,l}(x)\right] \\ &= \frac{1}{1-\alpha^*}\mathbb{E}_{\xi_x}\left[l(x,\xi_x)\mathbb{1}\left\{l(x,\xi_x) \geq V_{\alpha^*,l}(x)\right\}\right] \\ &= V_{\alpha^*,l}(x) + \frac{1}{1-\alpha^*}\mathbb{E}_{\xi_x}\left[\left(l(x,\xi_x) - V_{\alpha^*,l}(x)\right)^+\right], \end{aligned} \tag{2}$$

where $\mathbb{1}\{A\}$ is 1 if event A is true and 0 otherwise, $(u)^+ = \max(u,0)$, and $V_{\alpha^*,l}(x)$ is the VaR of $l(x,\xi_x)$ at risk level α^*, i.e.,

$$V_{\alpha^*,l}(x) = VaR_{\alpha^*}\left(l(x,\xi_x)\right) \stackrel{\triangle}{=} \inf\{t : P(t;x) \geq \alpha^*\} = P^{-1}(\alpha^*;x). \tag{3}$$

Note that the inverse c.d.f. $P^{-1}(\alpha^*;x)$ exists because $P(t;x)$ is strictly increasing in t.

Problem (1) might be difficult to solve when $l(x;\xi_x)$ lacks structural properties such as convexity and differentiability. Most of the gradient-based algorithms might fail. Instead, we seek model-based methods to solve problem (1). In principle, we could extend GASS algorithm in Zhou and Hu (2014) to solve the CVaR optimization problem (1).

2.1 Main Idea

Similar to many other model-based methods, the main idea of GASS is to introduce a parameterized sampling distribution over the solution space, and update the parameters of the sampling distribution iteratively towards the promising region of the solution space. Let us illustrate the main idea in a general framework, where one aims to maximize a deterministic function $L(x)$ over $x \in \mathscr{X}$.

We introduce a parameterized family of densities $\{f(x;\theta) : \theta \in \Theta \subset \mathbb{R}^{d_\theta}\}$ as the sampling distribution, where θ represents the parameter that will be updated over iterations. Consider a simple reformulation as follows:

$$H(\theta) \stackrel{\triangle}{=} \int L(x)f(x;\theta)dx.$$

Then $H(\theta) \leq L(x^*) = L^*$, where x^* denotes the optimal solution or one of the optima, and L^* denotes the optimal function value. Note that 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 reformulated problem $\max_{\theta\in\Theta} H(\theta)$ instead of the original problem, since the optimal parameter will recover the optimal solution and the optimal function value.

The advantage of the reformulated problem over the original problem is that it is differentiable in θ under mild regularity conditions on $f(x;\theta)$, and the gradient is easy to derive as follows:

$$\bigtriangledown_\theta H(\theta) = \bigtriangledown_\theta \int L(x)f(x;\theta)dx = \int L(x)\frac{\bigtriangledown_\theta f(x;\theta)}{f(x;\theta)}f(x;\theta)dx = \mathbb{E}_{f(\cdot;\theta)}\left[L(x)\bigtriangledown_\theta \ln f(x;\theta)\right].$$

Note that an unbiased estimator of $\bigtriangledown_\theta H(\theta)$ could be obtained by drawing samples $x^i \overset{i.i.d.}{\sim} f(x;\theta), i = 1,...,N$, evaluating $L(x^i)\bigtriangledown_\theta \ln f(x^i;\theta)$, and taking the sample average of $\{L(x^i)\bigtriangledown_\theta \ln f(x^i;\theta) : i = 1,...,N\}$. Therefore, one could solve the reformulated problem via a (stochastic) gradient-based method. Specifically, the method iteratively carries out the following two steps:

1. Generate candidate solutions according to the sampling distribution.
2. Based on the evaluation of the candidate solutions, update the parameter of the sampling distribution via gradient search.

Intuitively, it combines the relative fast convergence of gradient search with the robustness of model-based optimization in terms of maintaining a global exploration of the solution space.

2.2 Review of GASS

Based on the above main idea, now let us describe the full-blown GASS 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, and it satisfies the following conditions: for every θ, $S_\theta(y)$ is strictly 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 is to make the objective function positive while preserving the order of the solutions and in particular the optimal solution. Moreover, the shape function adds flexibility to the algorithm by giving a user the freedom to choose a weighting scheme on the samples based on sample function evaluations. For example, a good choice of the shape function $S_\theta(\cdot)$ is

$$S_\theta(L(x)) = \frac{1}{1+\exp(-S_0(L(x)-\gamma_\theta))}, \tag{4}$$

where S_0 is a large positive constant, and γ_θ is the $(1-\rho)$-quantile

$$\gamma_\theta \overset{\triangle}{=} \sup_r \left\{ r : P_{f(\cdot;\theta)}\left\{x \in \mathscr{X} : L(x) \geq r\right\} \geq \rho \right\}, \tag{5}$$

where $P_{f(\cdot;\theta)}\{A\}$ denotes the probability of event A w.r.t. $f(\cdot;\theta)$. Notice that $S_\theta(\cdot)$ could be viewed as a continuous approximation of the indicator function $\mathbb{1}\{L(x) \geq \gamma_\theta\}$ that eliminates the $L(x)$ values below γ_θ.

For an arbitrary but fixed $\theta' \in \Theta$, let us define

$$H(\theta;\theta') \overset{\triangle}{=} \int S_{\theta'}(L(x)) f(x;\theta) dx, \quad \text{and} \quad h(\theta;\theta') \overset{\triangle}{=} \ln H(\theta;\theta'). \tag{6}$$

By the condition on the shape function and the fact that $\ln(\cdot)$ is a strictly increasing function, solving the original problem is equivalent to solving the problem $\max_{\theta\in\Theta} h(\theta;\theta')$ for any fixed θ'. Following the main idea outlined before, Zhou and Hu (2014) propose a stochastic search algorithm that iteratively carries out the following two steps:

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

Note that the second step requires to compute the gradient and Hessian of $h(\theta;\theta_k)$, which, as shown by Zhou and Hu (2014), have analytical expressions as the expectations under certain probability measures. In particular, if the sampling distributions are chosen to be an exponential family of densities in the following Definition 1, then these expressions can be further simplified.

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

$$f(x;\theta) = \exp\left\{\theta^T \Gamma(x) - \eta(\theta)\right\}, \tag{7}$$

where $\Gamma(x) = [\Gamma_1(x), ..., \Gamma_d(x)]^T$ is the vector of sufficient statistics, $\eta(\theta) = \ln\{\int \exp(\theta^T \Gamma(x))dx\}$ is the normalization factor to ensure $f(x;\theta)$ is a p.d.f., and $\Theta = \{\theta : |\eta(\theta)| < \infty\}$ is the natural parameter space with a nonempty interior.

Proposition 1 below provides the corresponding analytical expressions of the gradient and Hessian of $h(\theta;\theta')$ when an exponential family of densities is used as the sampling distributions. We refer to Zhou and Hu (2014) for the detailed derivations.

Proposition 1 If $\{f(x;\theta) : \theta \in \Theta\}$ is an exponential family of densities, then the gradient and Hessian of $h(\theta;\theta')$ in (6) have closed-form expressions as follows:

$$\begin{cases} \bigtriangledown_\theta h(\theta;\theta')\big|_{\theta=\theta'} = \mathbb{E}_{q(\cdot;\theta')}[\Gamma(x)] - \mathbb{E}_{f(\cdot;\theta')}[\Gamma(x)], \\ \bigtriangledown^2_\theta h(\theta;\theta')\big|_{\theta=\theta'} = Var_{q(\cdot;\theta')}[\Gamma(x)] - Var_{f(\cdot;\theta')}[\Gamma(x)], \end{cases} \tag{8}$$

where

$$q(x;\theta') = \frac{S_{\theta'}(L(x))f(x;\theta')}{\int S_{\theta'}(L(x))f(x;\theta')dx}$$

is a "re-weighted" p.d.f., and $\mathbb{E}_{q(\cdot;\theta')}[\cdot]$ and $Var_{q(\cdot;\theta')}[\cdot]$ denote the expectation and variance w.r.t. $q(\cdot;\theta')$, respectively.

Note that the Hessian $\bigtriangledown^2_\theta h(\theta;\theta')\big|_{\theta=\theta'}$ might not be negative semi-definite. To ensure the parameter updating is along the ascent direction of $h(\theta;\theta')$, we approximate $\bigtriangledown^2_\theta h(\theta;\theta')\big|_{\theta=\theta'}$ by $-(Var_{\theta'}[T(x)] + \varepsilon I)$, which is a slight perturbation of the second term in $\bigtriangledown^2_\theta h(\theta;\theta')\big|_{\theta=\theta'}$ and negative definite. Here ε is a small positive number and I is an identity matrix of proper dimension. A Newton-like updating of θ is as follows:

$$\begin{aligned} \theta_{k+1} &= \Pi_\Theta\left\{\theta_k + \beta_k \left(Var_{f(\cdot;\theta_k)}[\Gamma(x)] + \varepsilon I\right)^{-1} \bigtriangledown_\theta h(\theta;\theta_k)\big|_{\theta=\theta_k}\right\} \\ &= \Pi_\Theta\left\{\theta_k + \beta_k \left(Var_{f(\cdot;\theta_k)}[\Gamma(x)] + \varepsilon I\right)^{-1} \left(\mathbb{E}_{q(\cdot;\theta_k)}[\Gamma(x)] - \mathbb{E}_{f(\cdot;\theta_k)}[\Gamma(x)]\right)\right\}, \end{aligned} \tag{9}$$

where β_k is a positive step-size, and $\Pi_\Theta\{\cdot\}$ denotes the projection operator that projects an iterate back onto the parameter space Θ by choosing the closest point in Θ.

In practical implementation, we still need to evaluate or estimate the expectation and variance terms in (9). Notice that the expectation term $\mathbb{E}_{f(\cdot;\theta_k)}[\Gamma(x)]$ can be calculated analytically in most cases. For example, if the chosen exponential family of densities is the Gaussian family, then $\mathbb{E}_{f(\cdot;\theta_k)}[\Gamma(x)]$ reduces to the mean and second moment of the Gaussian distribution. The variance term $Var_{f(\cdot;\theta_k)}[\Gamma(x)]$ might not be directly available, but it could be estimated by the sample variance using the candidate solutions drawn from $f(\cdot;\theta_k)$. Specifically, suppose N_k i.i.d. samples $\{x_k^i : i = 1, ..., N_k\}$ are drawn from $f(x;\theta_k)$, then

$$\frac{1}{N_k - 1}\sum_{i=1}^{N_k} \Gamma(x_k^i)\Gamma(x_k^i)^T - \frac{1}{N_k^2 - N_k}\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)^T$$

is the sample estimate of $Var_{f(\cdot;\theta_k)}[\Gamma(x)]$. The remaining term $\mathbb{E}_{q(\cdot;\theta_k)}[\Gamma(x)]$ can be estimated based on the principle of importance sampling, noting that $\mathbb{E}_{q(\cdot;\theta_k)}[\Gamma(x)] \propto \int S_{\theta_k}(L(x))\Gamma(x)f(x;\theta_k)dx$. Therefore, we could estimate it by $\sum_{i=1}^{N_k} w_k^i \Gamma(x_k^i)$, where $\{w_k^i : i = 1, ..., N_k\}$ are normalized weights given by

$$w_k^i = \frac{S_{\theta_k}(L(x_k^i))}{\sum_{j=1}^{N_k} S_{\theta_k}(L(x_k^j))}, \quad i = 1, ..., N_k.$$

2.3 Extension of GASS to Optimization of CVaR

When the CVaR of the loss function $C_{\alpha^*,l}(x)$ could be evaluated exactly for all $x \in \mathscr{X}$, we can directly extend the scheme described above to the CVaR minimization problem (1). Since the loss function is usually evaluated via simulation, its p.d.f. and c.d.f. are generally not available. Thus, the CVaR of the loss function, $C_{\alpha^*,l}(x)$, could not be evaluated analytically; however, it could be estimated via Monte Carlo simulation. In particular, suppose M samples of the loss function $\{l(x,\xi_x^1), l(x,\xi_x^2), ..., l(x,\xi_x^M)\}$ are simulated, and then sorted in ascending order as $l(x,\xi_x^{(1)}) \le l(x,\xi_x^{(2)}) \le ... \le l(x,\xi_x^{(M)})$, which forms the empirical distribution of the loss. A natural estimator of $C_{\alpha^*,l}(x)$ is the CVaR of the empirical distribution of the loss, which is as follows:

$$\widehat{C}_{\alpha^*,l}(x) \stackrel{\triangle}{=} \widehat{V}_{\alpha^*,l}(x) + \frac{1}{M(1-\alpha^*)} \sum_{m=1}^{M} \left(l(x,\xi_x^m) - \widehat{V}_{\alpha^*,l}(x) \right)^+, \tag{10}$$

where

$$\widehat{V}_{\alpha^*,l}(x) \stackrel{\triangle}{=} l\left(x, \xi_x^{(\lceil \alpha^* M \rceil)}\right) \tag{11}$$

is the VaR of the empirical distribution of the loss that plays the role of VaR estimator, and $\lceil \alpha^* M \rceil$ is the smallest integer that is greater than or equal to $\alpha^* M$.

Although the estimator $\widehat{C}_{\alpha^*,l}(x)$ is biased, it is strongly consistent under mild regularity assumptions on the distribution of $l(x,\xi_x)$ (see, e.g., Trindade et al. (2007)). In principle, we can use it as a replacement for $C_{\alpha^*,l}(x)$ and plug it into GASS algorithm.

3 ALGORITHMS: GASS-CVAR, GASS-CVAR-ARL

Now let us formally present the following algorithm, which is referred to as GASS-CVaR, for simulation optimization of CVaR.

Algorithm 1 **G**radient-based **A**daptive **S**tochastic **S**earch for Optimization of **CVaR**

1. **Initialization**: Choose an exponential family of densities $\{f(x;\theta) : \theta \in \Theta\}$, and specify a small positive constant ε, initial parameter θ_0, sample size sequence $\{N_k\}$ that satisfies $N_k \to \infty$, simulation budget sequence $\{M_k\}$, and step size sequence $\{\beta_k\}$ that satisfies $\sum_{k=0}^{\infty} \beta_k = \infty, \sum_{k=0}^{\infty} \beta_k^2 < \infty$. Set $k = 0$.
2. **Sampling**: Draw samples $\{x_k^i \stackrel{i.i.d.}{\sim} f(x;\theta_k) : i = 1,2,...,N_k\}$. For each x_k^i, evaluate the loss function scenarios $\{l(x_k^i, \xi_k^{i,j}) : j = 1,...,M_k\}$ by simulation, and sort them in ascending order, denoted by

$$l\left(x_k^i, \xi_k^{i,(1)}\right) \le l\left(x_k^i, \xi_k^{i,(2)}\right) \le \cdots \le l\left(x_k^i, \xi_k^{i,(M_k)}\right).$$

   Compute the corresponding CVaR estimate at target risk level α^*:

$$\widehat{C}_{\alpha^*,l}(x_k^i) = l\left(x_k^i, \xi_k^{i,(\lceil \alpha^* M_k \rceil)}\right) + \frac{1}{M_k(1-\alpha^*)} \sum_{j=1}^{M_k} \left(l\left(x_k^i, \xi_k^{i,j}\right) - l\left(x_k^i, \xi_k^{i,(\lceil \alpha^* M_k \rceil)}\right) \right)^+.$$

3. **Estimation**: Compute the normalized weights $\widehat{w}_k^i$ according to

$$\widehat{w}_k^i = \frac{S_{\theta_k}\left(-\widehat{C}_{\alpha^*,l}(x_k^i)\right)}{\sum_{j=1}^{N_k} S_{\theta_k}\left(-\widehat{C}_{\alpha^*,l}(x_k^j)\right)}, \quad i = 1,...,N_k,$$

and then $\mathbb{E}_{q(\cdot;\theta_k)}[\Gamma(x)]$ and $Var_{f(\cdot;\theta_k)}[\Gamma(x)]$ are estimated via

$$\begin{cases} \widehat{\mathbb{E}}_{q(\cdot;\theta_k)}[\Gamma(x)] = \sum_{i=1}^{N_k} \widehat{w}_k^i \Gamma(x_k^i), \\ \widehat{Var}_{f(\cdot;\theta_k)}[\Gamma(x)] = \frac{1}{N_k-1}\sum_{i=1}^{N_k} \Gamma(x_k^i)\Gamma(x_k^i)^T - \frac{1}{N_k^2-N_k}\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)^T. \end{cases}$$

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

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}}\left\{\theta_k + \beta_k\left(\widehat{Var}_{f(\cdot;\theta_k)}[\Gamma(x)] + \varepsilon I\right)^{-1}\left(\widehat{\mathbb{E}}_{q(\cdot;\theta_k)}[\Gamma(x)] - \mathbb{E}_{f(\cdot;\theta_k)}[\Gamma(x)]\right)\right\},$$

where $\widetilde{\Theta} \subseteq \Theta$ is a non-empty compact and convex constraint set.

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) of GASS-CVaR (Algorithm 1), the conditions on the sample size and step size sequences are imposed to facilitate the convergence of the algorithm. In the sampling step (step 2), notice that the CVaR estimates are biased. Therefore, the convergence of the original GASS algorithm does not directly apply to GASS-CVaR. We leave the convergence analysis of GASS-CVaR to future work. In the estimating step (step 3), as mentioned before, one common choice of the shape function $S_\theta(\cdot)$ is in the form of expression (4). Moreover, the quantile level ρ in (5) controls the percentile of elite samples that are used to update the sampling distribution at the next iteration, and balances the exploitation of the neighborhood of the current best solutions with the exploration of the entire solution space. For example, when a smaller ρ is used, less elite samples are used, and thus less emphasis is put on exploration. In the updating step (step 4), the iterate is projected onto a convex and compact subset $\widetilde{\Theta} \subseteq \Theta$ instead of Θ, in order to guarantee numerical stability and fast computation of the projection. In the stopping step (step 5), a common stopping criterion used in practice is that the norm of the gradient falls below a pre-specified threshold.

3.1 GASS with Adaptive Risk Level

When the risk level of interest α^* is close to 1 (e.g. $\alpha^* = 0.99$), implementing GASS-CVaR could be computationally expensive, since the CVaR evaluation in step 2 requires a large simulation budget M_k to obtain good CVaR estimators. This issue is more severe as α^* gets closer to 1. For example, for a fixed x, suppose we want to estimate $C_{\alpha,l}(x)$ at three different risk levels as follows: $\alpha_1 = 0$, $\alpha_2 = 0.90$, and $\alpha_3 = 0.99$, where note that $C_{\alpha_1=0,l}(x) = \mathbb{E}[l(x,\xi_x)]$ is the expected loss. To achieve the same accuracy in estimation of CVaR, the corresponding simulation budgets M_1, M_2, and M_3 should result in equal "effective" simulation budgets $(1-\alpha_i)M_i, i = 1,2,3$. Therefore, $M_2 = (1-\alpha_1)/(1-\alpha_2)\cdot M_1 = 10\cdot M_1$ and $M_3 = (1-\alpha_1)/(1-\alpha_3)\cdot M_1 = 100\cdot M_1$. This implies that the simulation budget required for CVaR estimation could be easily up to tens of times even hundreds of times compared with the simulation budget required for the estimation of expectation.

To save simulation budget and improve the overall efficiency of GASS-CVaR, we propose to initialize the algorithm at a small risk level α_0 (e.g., $\alpha_0 = 0$), and adaptively increase/update the risk level α_k at every iteration until the target risk level α^* is achieved and the algorithm converges simultaneously. Since a lower risk level implies that a smaller simulation budget M_k is required to achieve desired accuracy for CVaR estimation, the hope is to adaptively save simulation budget at each iteration by solving a problem that is similar to the original one but less computationally expensive. A good updating rule on the risk level should 1) achieve significant budget savings when the algorithm is in the "warm-up" phase, i.e., when it puts more emphasis on exploration of the entire solution space; 2) solve problems that are close to the original one when the algorithm is in the "convergence" phase, i.e., when it put more emphasis on the exploitation of the promising region that has been identified, so that good solutions of the original problem

can be found. The key to such an updating rule lies in finding an empirical indication on the algorithm's balance between the exploration and the exploitation.

Note that GASS-CVaR maintains the structure of a gradient-based optimization scheme, and thus the gradient (even the Hessian) used in the updating step (step 4) could be regarded as the empirical indication on the balance between the exploration and the exploitation. It is natural to design the updating rule on the risk level using the information contained in the gradient obtained at every iteration. For example, note that GASS-CVaR converges when the norm of the gradient hits zero. Then one could increase the risk level at every iteration proportionally to the decrease in the norm of the gradient from the previous iteration. In particular, we propose the following version of GASS algorithm with adaptive risk levels, which is referred to as GASS-CVaR-ARL. We do point out that more sophisticated updating rules on the risk level could be incorporated in the future.

Algorithm 2 GASS-CVaR with **A**daptive **R**isk **L**evels

1. **Initialization**: Initialize the algorithm similar to step 1 in GASS-CVaR. Set initial risk level α_0.
2. Draw candidate solution samples and simulate the loss function scenarios similar to step 2 in GASS-CVaR. Compute the corresponding CVaR estimate at the risk level α_k:

$$\widehat{C}_{\alpha_k,l}(x_k^i) = l\left(x_k^i, \xi_k^{i,(\lceil \alpha_k M_k \rceil)}\right) + \frac{1}{M_k(1-\alpha_k)} \sum_{j=1}^{M_k} \left(l\left(x_k^i, \xi_k^{i,j}\right) - l\left(x_k^i, \xi_k^{i,(\lceil \alpha_k M_k \rceil)}\right)\right)^+.$$

   Record the best solution x_k^* found at this iteration: $x_k^* = \arg\min_i \widehat{C}_{\alpha_k,l}(x_k^i)$.
3. **Estimation**: Compute the normalized weights $\widehat{w}_k^i$ according to

$$\widehat{w}_k^i = \frac{S_{\theta_k}\left(-\widehat{C}_{\alpha_k,l}(x_k^i)\right)}{\sum_{j=1}^{N_k} S_{\theta_k}\left(-\widehat{C}_{\alpha_k,l}(x_k^j)\right)}, \quad i = 1,...,N_k,$$

   and then $\mathbb{E}_{q(\cdot;\theta_k)}[\Gamma(x)]$ and $Var_{f(\cdot;\theta_k)}[\Gamma(x)]$ are estimated via

$$\begin{cases} \widehat{\mathbb{E}}_{q(\cdot;\theta_k)}[\Gamma(x)] = \sum_{i=1}^{N_k} \widehat{w}_k^i \Gamma(x_k^i), \\ \widehat{Var}_{f(\cdot;\theta_k)}[\Gamma(x)] = \frac{1}{N_k-1}\sum_{i=1}^{N_k} \Gamma(x_k^i)\Gamma(x_k^i)^T - \frac{1}{N_k^2-N_k}\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)\left(\sum_{i=1}^{N_k}\Gamma(x_k^i)\right)^T. \end{cases}$$

   Compute the gradient g_k via

$$g_k = \widehat{\mathbb{E}}_{q(\cdot;\theta_k)}[\Gamma(x)] - \mathbb{E}_{f(\cdot;\theta_k)}[\Gamma(x)].$$

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

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}}\left\{\theta_k + \beta_k\left(\widehat{Var}_{f(\cdot;\theta_k)}[\Gamma(x)] + \varepsilon I\right)^{-1} g_k\right\},$$

   where $\widetilde{\Theta} \subseteq \Theta$ is a non-empty compact and convex constraint set, and update the risk level α according to

$$\alpha_{k+1} = \begin{cases} \alpha^* - \frac{\|g_k\|_2}{\|g_{k-1}\|_2}(\alpha^* - \alpha_k), & \text{if } \|g_k\|_2 < \|g_{k-1}\|_2, \\ \alpha_k, & o/w, \end{cases} \tag{12}$$

   where $\|\cdot\|_2$ is the Euclidean norm.
5. **Stopping**: Check if some stopping criterion is satisfied. If yes, stop and return $x^* = \arg\min_k \widehat{C}_{\alpha^*,l}(x_k^*)$ and $\widehat{C}_{\alpha^*,l}(x^*)$ via simulation; else, set $k := k+1$ and go back to step 2.

In the sampling step (step 2) of GASS-CVaR-ARL (Algorithm 2), since the current risk level α_k is smaller than the target risk level α^*, we could use a simulation budget M_k that is smaller than the one used in GASS-CVaR to estimate the CVaR at risk level α_k. For example, suppose one wants to keep the "effective" simulation budget $(1-\alpha_k)M_k$ as a constant. Then, in the initial iterations of the algorithm the budget savings can be up to tens of times even hundreds of times (equal to $(1-\alpha_k)/(1-\alpha^*)$ precisely) since α_k is close to $\alpha_0 = 0$. The sampled best solution to the CVaR optimization problem at risk level α_k is also recorded. It can be viewed as a good solution to a CVaR optimization problem that is similar to the original one. In the updating step (step 4), the updating rule (12) ensures that α_k is non-decreasing, with the hope that α_k will eventually converge to the target risk level α^*. Furthermore, when $\|g_k\|_2 < \|g_{k-1}\|_2$, we can rewrite (12) as $\frac{\alpha^*-\alpha_{k+1}}{\alpha^*-\alpha_k} = \frac{\|g_k\|_2}{\|g_{k-1}\|_2}$. Loosely speaking, it implies the increase in risk level for the next iteration is proportional to the decrease in the norm of the gradient from the previous iteration. It also ensures that the target risk level is achieved when the norm of the gradient hits zero, i.e., when the algorithm converges. In the stopping step (step 5), finding the best solution to the original CVaR optimization problem is achieved via evaluating and comparing the CVaR values at the target risk level for all the best sampled solutions found so far, and thus additional simulation budget is required; however, it is insignificant compared with the overall budget consumed. We leave the convergence analysis of GASS-CVaR-ARL to future work.

Recall that, in GASS-CVaR-ARL, the risk level α_k is updated in accordance with the decrease in the norm of the gradient. It implies that the updating rule (12) keeps track of the algorithm's balance between the exploration and the exploitation, and makes adjustments on the risk level accordingly. Therefore, in the "warm-up" phase of the algorithm, having a small α has little negative effect on the algorithm progress since the algorithm put most of its emphasis on exploration; in the "convergence" phase of the algorithm, the risk level α_k is close to the target risk level α^*, and essentially the algorithm is solving problems close to the original one. Thus, intuitively, we expect that the number of iterations that GASS-CVaR-ARL takes to converge is similar to the one that GASS-CVaR takes to converge, which is also verified by the numerical tests (presented in the next section). Since GASS-CVaR-ARL saves simulation budget at every iteration, overall budget saving is achieved.

4 NUMERICAL EXPERIMENTS

We carry out numerical tests to compare the performance of GASS-CVaR and GASS-CVaR-ARL. In particular, the loss functions tested are listed in the following, among which some are designed by adding Gaussian noises to the continuous benchmark functions in Hu et al. (2007). However, we point out our algorithms do not have much assumption on the structure of the loss function as well as the noise. For convenience, let $\mathcal{N}(0,1)$ be a standard one-dimensional Gaussian distribution, and the loss function is in the form of

$$l_i(x,\xi_x) = L_i(x) + \begin{cases} \sqrt{1+100\sum_{d=1}^{D}(x_d-1)^2}\cdot\mathcal{N}(0,1),\ i=0,1,3,4, \\ \sqrt{1+100\sum_{d=1}^{D}(x_d-2)^2}\cdot\mathcal{N}(0,1),\ i=2,5, \end{cases} \tag{13}$$

where D is the dimension of the solution space. Specifically, $L_0 = \sum_{d=1}^{D} x_d^2$; L_1 and L_2 are respectively Powell function and Rosenbrock function, which are badly scaled; L_3 is Rastrigin function, which is multimodal with a large number of local optima; L_4 and L_5 are respectively Pintér function and Levy function, which are badly-scaled as well as multimodal. We test all functions with $D = 10$.

Notice that if $C_{\alpha^*=0,l_i}(x) = L_i(x) = \mathbb{E}_{\xi_x}[l_i(x,\xi_x)]$ is of interest, then evidently $x^\circ = [0,...,0]_d$ is the minimizer for $i = 0,1,3,4$, and $x^\circ = [1,...,1]_d$ is the minimizer for $i = 2,5$. As the risk level of interest α^* increases, the minimizer of $C_{\alpha^*,l_i}(x)$, might be very different from x°, since the loss is exposed to a lot noise at x°. In fact, the minimizer of $C_{\alpha^*,l_i}(x)$ will move towards the region where the mean loss and the noise of the loss are balanced. In particular, for $i = 0,1,3,4$, it will move towards $x = [1,...,1]_d$, where the loss has the least noise; for $i = 2,5$, it will move towards $x = [2,...,2]_d$. Note that, except for the loss function

l_0, the minimizers of $C_{\alpha^*,l_i}(x)$ and the minimum CVaR function values are not analytically available for any other loss function listed in (13) when $\alpha^* > 0$.

In all the implementations, we use independent multivariate normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$ as the parameterized sampling distribution $f(x;\theta_k)$ at iteration k, where $\mu_k = (\mu_k^1, ..., \mu_k^D)^T$ is the mean parameter and $\Sigma_k = diag((\sigma_k^1)^2, ..., (\sigma_k^D)^2)$ is the covariance matrix. Thus, $\theta_k = (\mu_k^1, ..., \mu_k^D; (\sigma_k^1)^2, ..., (\sigma_k^D)^2)^T$. The initial mean parameter μ_0 are drawn randomly from the uniform distribution $U[-30, 30]^D$, and the initial covariance matrix Σ_0 is set to be $\Sigma_0 = 1000 I_{D\times D}$, where $I_{D\times D}$ is the identity matrix of dimension D. From the experiment results, we notice that the performance of the algorithms is insensitive to the initial mean parameter as long as the initial covariance matrix is sufficiently large.

At iteration k, we use the shape function $S_{\theta_k}(\cdot)$ in the form of expression (4) with $S_0 = 10^5$ and $\rho = 0.1$ in (5). The $(1-\rho)$-quantile γ_{θ_k} is estimated by the $(1-\rho)$ sample quantile of the CVaR estimates for all the candidate solutions generated at this iteration. The risk level of interest is $\alpha^* = 0.99$, and in GASS-CVaR-ARL the initial risk level is set to be $\alpha_0 = 0$. The sample size of candidate solutions drawn from the sampling distribution is set to be $N_k = 1000$, and the simulation budget used to estimate the CVaR of the loss function is set in a way such that the effective simulation budget is $(1-\alpha_k)M_k = 50$. Therefore, in GASS-CVaR $M_k = 50/(1-\alpha^*) = 50/0.01 = 5\times 10^3$ for all k, and in GASS-CVaR-ARL $M_k = 50/(1-\alpha_k)$ at iteration k with initial simulation budget $M_0 = 50/(1-0) = 50$. The small positive constant ε used to ensure the positive definiteness of the Hessian is set to be $\varepsilon = 10^{-10}$, and the step size β_k is set to be $\beta_k = 50/(k+2000)^{0.6}$, which satisfies the assumptions in step 1 of both two algorithms.

We run both algorithms 50 times independently and summarize their average performance in Figure 1. Recall that, except for the loss function l_0, the minimum CVaR value is not readily available for any other loss function. So we implement GASS-CVaR with constant sample size $N = 10^3$ and simulation budget $M = 10^5$ to find them. In the upper-left plot for the loss function l_0 in Figure 1, the y-axis represents the ratio of CVaR values obtained by the algorithms to the true minimum CVaR value at the target risk level α^*; for all the rest of the plots, the y-axis represents the same ratio, except that the true minimum is replaced by the smallest CVaR values obtained from implementing GASS-CVaR with sample size N and simulation budget M. We observe that both algorithms (GASS-CVaR and GASS-CVaR-ARL) perform well in finding optimal solutions and minimum CVaR values. Moreover, GASS-CVaR-ARL converges faster and often reduces the total number of function evaluations needed for convergence by 2-4 times, which demonstrates the advantage of using adaptive risk level in GASS-CVaR-ARL.

Figure 2 includes two plots for the loss function l_0: the left one plots the ratio of the CVaR values evaluated at the means of the sampling distributions to the true minimum CVaR value (recall that it is available for l_0); the right one plots the trajectory of the risk level α_k. We can see that the means of the sampling distributions in both GASS-CVaR and GASS-CVaR-ARL converge to the optimal solution, and GASS-CVaR-ARL achieves a faster convergence speed. Moreover, the risk level α_k in GASS-CVaR-ARL increases steadily to the target risk level α^*, which indicates that the norm of the gradient decreases steadily to zero and the algorithm converges.

5 CONCLUSION

In this paper, we study the application of the recently proposed GASS algorithm for deterministic non-differentiable optimization to the optimization of CVaR. Instead of optimizing CVaR at the risk level of interest directly, we propose to initialize the algorithm at a small risk level, and then increase the risk level at each iteration adaptively until the target risk level is achieved, while the algorithm converges to an optimal solution of the original problem. It enables us to adaptively reduce the number of samples needed to estimate CVaR at each iteration, and eventually improves the efficiency. The numerical results demonstrate the advantage of incorporating such an adaptive updating rule on the risk level in the algorithm by showing it achieves 2-4 times of budget savings for the tested loss functions.

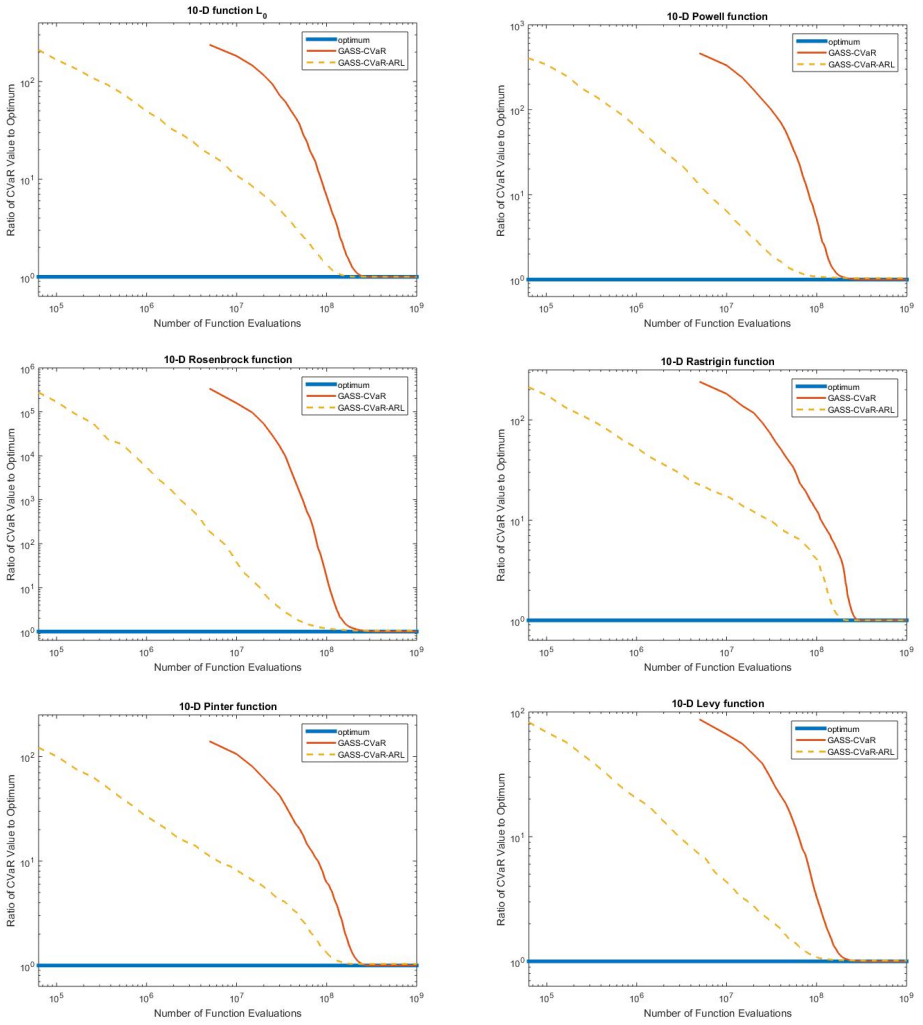

Figure 1: Average Performance of GASS-CVaR and GASS-CVaR-ARL.

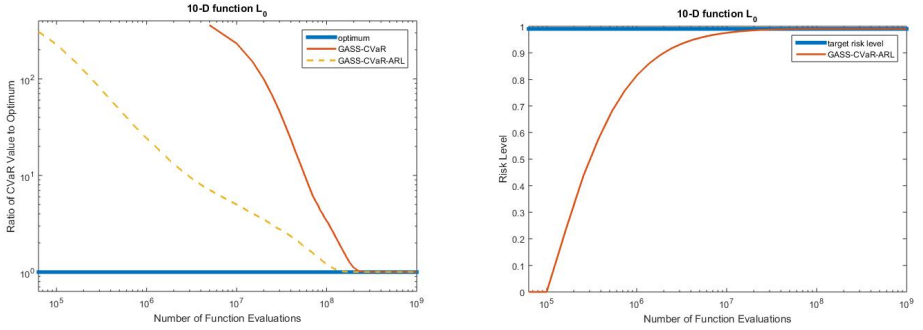

Figure 2: CVaR at Mean of the Sampling Distribution and Trajectory of Risk Level.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation under Grants CMMI-1413790 and CAREER CMMI-1453934, and Air Force Office of Scientific Research under Grant YIP FA-9550-14-1-0059.

REFERENCES

Alexander, S., T. F. Coleman, and Y. Li. 2006. "Minimizing CVaR and VaR for A Portfolio of Derivatives". *Journal of Banking & Finance* 30 (2): 583–605.

Artzner, P., F. Delbaen, J.-M. Eber, and D. Heath. 1999. "Coherent Measures of Risk". *Mathematical Finance* 9:203–228.

Hong, L. J. 2009. "Estimating Quantile Sensitivities". *Operations Research* 57 (1): 118–130.

Hong, L. J., and G. Liu. 2009. "Simulating Sensitivities of Conditional Value at Risk". *Management Science* 55 (2): 281–293.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. "A Model Reference Adaptive Search Method for Global Optimization". *Operations Research* 55 (3): 549–568.

Rockafellar, R. T., and S. Uryasev. 2000. "Optimization of Conditional Value-at-Risk". *Journal of Risk* 2:21–42.

Rockafellar, R. T., and S. Uryasev. 2002. "Conditional Value-at-Risk for General Loss Distributions". *Journal of Banking & Finance* 26 (7): 1443–1471.

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, Cross-Entropy, Ants and Rare Events". *Stochastic Optimization: Algorithms and Applications* 54:303–363.

Trindade, A. A., S. Uryasev, A. Shapiro, and G. Zrazhevsky. 2007. "Financial Prediction with Constrained Tail Risk". *Journal of Banking & Finance* 31 (11): 3524–3538.

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

HELIN ZHU is a Ph.D. student in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. His research interests include stochastic control and simulation optimization. His email address is hzhu67@gatech.edu.

JOSHUA HALE is a Ph.D. student in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology. His research interests include simulation optimization and its applications. His email address is jhale32@gatech.edu.

ENLU ZHOU is an Assistant Professor in the H. Milton Stewart School of Industrial and 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 received 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 e-mail address is enlu.zhou@isye.gatech.edu and her web page is http://enluzhou.gatech.edu/.