Simulation Optimization of Risk Measures with Adaptive Risk Levels

Helin Zhu, Joshua Hale, and Enlu Zhou H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology

Abstract

Optimizing risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) of a general loss distribution is usually difficult, because 1) the loss function might lack structural properties such as convexity or differentiability since it is often generated via black-box simulation of a stochastic system; 2) evaluation of risk measures often 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) to the optimization of risk measures VaR and CVaR. Instead of optimizing VaR or CVaR at the target risk level directly, we incorporate an adaptive updating scheme on the risk level, by initializing the algorithm at a small risk level and adaptively increasing it until the target risk level is achieved while the algorithm converges at the same time. This enables us to adaptively reduce the number of samples required to estimate the risk measure at each iteration, and thus improving the overall efficiency of the algorithm.

Key words: Risk measures, black-box simulation, rare-event simulation, GASS, adaptive risk level

1 Introduction

Risk measures such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) are widely studied in various fields, in order to quantify the extreme behaviors of the distributions of interest. Loosely speaking, VaR characterizes a tail quantile of a distribution, and CVaR characterizes the conditional expectation of the tail portion of the 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. An abundant literature has dedicated to studying the estimation and optimization of risk measures under various settings. Rockafellar and Uryasev (2000) and Rockafellar and Uryasev (2002) derive some fundamental properties of CVaR for general loss distributions in finance, and propose the fundamental minimization formula to facilitate the optimization of general coherent risk measures, and derive the optimality conditions via the dual representation. Ruszczyński (2010) study the optimization of risk measures under a multistage setting, and propose a risk-averse dynamic programming approach to risk optimization in Markov decision processes. Alexander et al. (2006) study the optimization of VaR and CVaR for derivatives portfolios with the addition of a cost that is proportional to the portfolio position.

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 available 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 requirements on problem structure. Examples of model-based methods include but are not limited to ant colony optimization (Dorigo and Blum (2005)), annealing adaptive search (AAS) (Romeijn and Smith (1994)), the estimation of distribution algorithms (EDA) (Larranaga and Lozano (2002)), the cross-entropy (CE) method (Rubinstein (2001)), model reference adaptive search (MRAS) (Hu et al. (2007) and Hu et al. (2008)), the interacting-particle algorithm (Molvalioglu et al. (2009), Molvalioglu et al. (2010)), and gradient-based adaptive stochastic search (GASS) (Zhou and Hu (2014)).

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 CE and MRAS, the updating rule is derived by minimizing the Kullback-Leibler (K-L) divergences between a converging sequence of reference distributions and a chosen exponential family of densities. For another example, in 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 a Newton-like scheme.

Compared with traditional 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 exploration and exploitation.

Although all the aforementioned model-based methods are designed for deterministic optimization problems, they can be extended to risk (VaR or CVaR) optimization problems in which the exact risk values are replaced with (biased) sample estimates. However, a straightforward extension usually leads to a computationally expensive algorithm, due to the rare-event simulation required in estimating the risk values. This issue is even more severe when the risk level of interest is close to 1, which is often the case for risk management practitioners. It inspires us to consider the following question. For a risk optimization problem, is it possible to initialize a model-based algorithm at a small risk level (close to 0), and then adaptively increase the risk level at every iteration such that the target risk

level is achieved while the algorithm converges at the same time? The motivation is that the algorithm will consume less simulation budget (since the risk level is small) during the "warm-up" phase of the algorithm, and solve problems that are close to the original one during the "convergence" phase of the algorithm, eventually leading to total budget saving. The key to this question lies in finding a signal to link the updating rule on the risk level with the updating rule on the sampling distribution parameter, where the signal is capable of measuring empirically the algorithm's emphasis between the exploitation of a promising region and the exploration of the entire solution space.

In this paper, we will focus on the extension of a specific model-based methods, i.e., 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) in the updating rule of the sampling distribution parameter can be viewed as a signal that empirically measures the algorithm's emphasis between the exploitation of a promising region and the exploration of the entire solution space. 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. We will show that incorporating such an updating scheme on the risk level in the algorithm guarantees that the target risk level is achieved at the same time when the algorithm converges. Furthermore, compared with vanilla GASS, we will show that the proposed algorithm achieves significant total budget savings.

To the best of our knowledge, this work is among the first to apply a model-based algorithm to risk optimization problems, and among the first to propose a risk optimization scheme with adaptive risk levels. To ease the presentation, 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.

The rest of the paper is organized as follows. In Section 2, we will first describe the CVaR optimization problem. Then we extend the algorithm GASS, 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 that adaptively adjusts the risk level. Convergence analysis of both algorithms are presented in Section 4. In Section 5, we illustrate the performance of the proposed algorithms by carrying out numerical tests on several benchmark loss functions. We conclude the paper in Section 6.

2 General Framework

Consider a scalar loss function of the form $l(x, \xi_x)$, where $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ represents the decision variable, and ξ_x represents the randomness in the loss function. The distribution of $l(x, \xi_x)$ 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, we 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 $x \in \mathcal{X}$. The objective is to minimize CVaR of $l(x; \xi_x)$ at a risk level of interest α^* $(0 < \alpha^* < 1)$ with respect to (w.r.t.) $x \in \mathcal{X}$. That is, to solve the following stochastic optimization problem:

$$\min_{x \in \mathcal{X}} C_{\alpha^*}(x) \stackrel{\triangle}{=} CVaR_{\alpha^*}(l(x,\xi_x)), \quad \text{or equivalently}, \quad \max_{x \in \mathcal{X}} -C_{\alpha^*}(x), \quad (2.1)$$

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

$$CVaR_{\alpha^*}(l(x,\xi_x)) \stackrel{\Delta}{=} \mathbb{E}_{\xi_x}[l(x,\xi_x)|l(x,\xi_x) \ge V_{\alpha^*}(x)]$$

= $\frac{1}{1-\alpha^*} \mathbb{E}_{\xi_x}[l(x,\xi_x)\mathbb{1}\left\{l(x,\xi_x) \ge V_{\alpha^*}(x)\right\}]$
= $V_{\alpha^*}(x) + \frac{1}{1-\alpha^*} \mathbb{E}_{\xi_x}\left[\left(l(x,\xi_x) - V_{\alpha^*}(x)\right)^+\right],$

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

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

Note that the inverse c.d.f. $P^{-1}(\alpha^*; x)$ exists because P(t; x) is strictly increasing in t. We also follow the standard assumption that $C_{\alpha^*}(x)$ is bounded from below and above on \mathcal{X} , i.e., $\exists C_{lb} > -\infty, C_{ub} < \infty$ s.t. $C_{lb} < C_{\alpha^*}(x) < C_{ub}, \forall x \in \mathcal{X}$.

Problem (2.1) might be difficult to solve when $l(x; \xi_x)$ lacks structural properties such as convexity and differentiability. Thus, traditional gradient-based method might not be applicable. In contrast, model-based methods are good alternatives as in general they impose few requirements on the structure of $l(x; \xi_x)$. Therefore, we will apply a model-based method to solve problem (2.1). In principle, we could extend the algorithm GASS in Zhou and Hu (2014), which is reviewed in next section.

2.1 Review of GASS

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 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 $x \in \mathcal{X}$.

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:

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

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 on \mathcal{X} . 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 , one could 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.

An 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:

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

Note that an unbiased estimator of $\nabla_{\theta} H(\theta)$ could be obtained by drawing independent and identically distributed (i.i.d.) samples $x^i \sim f(x;\theta), i = 1, ..., N$, evaluating $L(x^i) \nabla_{\theta}$ $\ln f(x^i;\theta)$, and taking the sample average of $\{L(x^i) \nabla_{\theta} \ln f(x^i;\theta)\}$. 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.

Based on the above main idea, now let us review the full-blown GASS algorithm. 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 shape function S_{θ} satisfies the following conditions: for every θ , $S_{\theta}(y)$ is strictly increasing in y, and bounded below from zero and above for finite y; moreover, for every fixed y, $S_{\theta}(y)$ is continuous in θ . The purpose of introducing S_{θ} is to make the objective function positive, and yet preserve the order of the solutions and in particular the optimal solution. Moreover, the dependence of S_{θ} on θ adds flexibility to the algorithm by giving user the freedom to choose a weighting scheme on the samples based on the function evaluations. For example, a good choice of $S_{\theta}(\cdot)$ is

$$S_{\theta}(L(x)) = \frac{1}{1 + \exp(-S_o(L(x) - \gamma_{\theta}))},$$
(2.2)

where S_o is a large positive constant, and γ_{θ} is the $(1 - \rho)$ -quantile

$$\gamma_{\theta} \stackrel{\Delta}{=} \sup_{r} \left\{ r : P_{f(\cdot;\theta)} \left\{ x \in \mathcal{X} : L(x) \ge r \right\} \ge \rho \right\},$$
(2.3)

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 gives equal weights to the solutions with function values above γ_{θ} and eliminates the solutions with function values below γ_{θ} .

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

$$H(\theta;\theta') \stackrel{\triangle}{=} \int S_{\theta'}(L(x))f(x;\theta)dx, \quad \text{and} \quad h(\theta;\theta') \stackrel{\triangle}{=} \ln H(\theta;\theta').$$
(2.4)

By the condition on the shape function and the fact that $\ln(\cdot)$ is a strictly increasing function, the original problem can be transform to $\max_{\theta \in \Theta} h(\theta; \theta')$ 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 sampling distribution parameter obtained at iteration k.
- 2. Update $\theta_k \to \theta_{k+1}$ using a Newton-like iteration for $\max_{\theta} h(\theta; \theta_k)$, where $h(\theta; \theta_k) = \ln\{\int S_{\theta_k}(L(x))f(x; \theta)dx\}$.

Note that the second step requires us to compute the gradient and Hessian of $h(\theta; \theta_k)$ at $\theta = \theta_k$, which, as shown by Zhou and Hu (2014), have analytical expressions as expectations under certain probability measures. In particular, if the sampling distribution belongs to an exponential family of densities, which is defined in the following Definition 2.1, then these expressions can be further simplified.

Definition 2.1. A family $\{f(x; \theta) : \theta \in \Theta\}$ is an exponential family of densities if it satisfies

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

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 that ensures $f(x;\theta)$ to be a p.d.f.; $\Theta = \{\theta : |\eta(\theta)| < \infty\}$ is the natural parameter space with a nonempty interior.

Proposition 2.1 below provides the corresponding analytical expressions of the gradient and Hessian of $h(\theta; \theta_k)$ at $\theta = \theta_k$, when an exponential family of densities is used as the sampling distribution. We refer to Zhou and Hu (2014) for the detailed derivations.

Proposition 2.1. If $\{f(x;\theta) : \theta \in \Theta\}$ is an exponential family of densities, then the gradient and Hessian of $h(\theta; \theta_k)$ at $\theta = \theta_k$ have closed-form expressions as follows.

$$\begin{cases} \nabla_{\theta} h(\theta; \theta_k) \big|_{\theta = \theta_k} = \mathbb{E}_{q(\cdot; \theta_k)} \left[\Gamma(x) \right] - \mathbb{E}_{\theta_k} \left[\Gamma(x) \right], \\ \nabla_{\theta}^2 h(\theta; \theta_k) \big|_{\theta = \theta_k} = Var_{q(\cdot; \theta_k)} \left[\Gamma(x) \right] - Var_{\theta_k} \left[\Gamma(x) \right], \end{cases}$$
(2.5)

where

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

is a "re-weighted" p.d.f.; $\mathbb{E}_{q(\cdot;\theta_k)}[\cdot]$ and $Var_{q(\cdot;\theta_k)}[\cdot]$ denote the expectation and variance w.r.t. $q(\cdot;\theta_k)$, respectively; $\mathbb{E}_{\theta_k}[\cdot]$ and $Var_{\theta_k}[\cdot]$ denote the expectation and variance w.r.t. $f(\cdot;\theta_k)$, respectively.

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

$$\theta_{k+1} = \Pi_{\Theta} \left\{ \theta_k + \beta_k \left(Var_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \bigtriangledown_{\theta} h(\theta; \theta_k) \Big|_{\theta = \theta_k} \right\}$$

$$= \Pi_{\Theta} \left\{ \theta_k + \beta_k \left(Var_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \left(\mathbb{E}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)] \right) \right\},$$
(2.6)

where β_k is a positive step-size, $\mathbb{E}_{q_k}[\cdot]$ denotes the expectation w.r.t. $q(\cdot; \theta_k)$, 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 .

To have an implementable algorithm, the expectation and variance terms in (2.6) need to be evaluated or estimated. Notice that the expectation term $\mathbb{E}_{\theta_k}[\Gamma(x)]$ can be calculated analytically in most cases. For example, if the chosen exponential family of densities is the Gaussian family, then $\mathbb{E}_{\theta_k}[\Gamma(x)]$ reduces to the mean and second moment of a Gaussian distribution. The variance term $Var_{\theta_k}[\Gamma(x)]$ 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

$$\widehat{Var}_{\theta_k}\left[\Gamma(x)\right] \stackrel{\triangle}{=} \frac{1}{N_k - 1} \sum_{i=1}^{N_k} \Gamma(x_k^i) \Gamma(x_k^i)^T - \frac{1}{N_k^2 - N_k} \left(\sum_{i=1}^{N_k} \Gamma(x_k^i)\right) \left(\sum_{i=1}^{N_k} \Gamma(x_k^i)\right)^T$$
(2.7)

is a sample estimate of $Var_{\theta_k}[\Gamma(x)]$. The remaining term $\mathbb{E}_{q_k}[\Gamma(x)]$ can be estimated using the principle of importance sampling with samples $\{x_k^i\}$, noting that

$$\mathbb{E}_{q_k}\left[\Gamma(x)\right] \propto \int S_{\theta_k}(L(x))\Gamma(x)f(x;\theta_k)dx$$

That is, the expectation $\mathbb{E}_{q_k}[\Gamma(x)]$ could be estimated by

$$\widetilde{\mathbb{E}}_{q_k}\left[\Gamma(x)\right] \stackrel{\triangle}{=} \sum_{i=1}^{N_k} w_k^i \Gamma(x_k^i), \qquad (2.8)$$

where $\{w_k^i : i = 1, ..., N_k\}$ are self-normalized weights given by

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

When $S_{\theta_k}(\cdot)$ takes a form such as (2.2), it has to be estimated by samples as well since the $(1 - \rho)$ -quantile γ_{θ_k} defined in (2.3) needs to be estimated by a sample $(1 - \rho)$ -quantile. Denote the sample quantile by $\hat{\gamma}_{\theta_k}$ and the resulted approximate shape function by $\hat{S}_{\theta_k}(\cdot)$. Then, $\{w_k^i\}$ are approximated according to

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

and thus $\mathbb{E}_{q_k}[\Gamma(x)]$ is approximated by

$$\widehat{\mathbb{E}}_{q_k}\left[\Gamma(x)\right] \stackrel{\triangle}{=} \sum_{i=1}^{N_k} \widehat{w}_k^i \Gamma(x_k^i).$$
(2.9)

Eventually, the gradient $g_k := \mathbb{E}_{q_k} [\Gamma(x)] - \mathbb{E}_{\theta_k} [\Gamma(x)]$ is approximated by

$$\widehat{g}_{k} \stackrel{\triangle}{=} \widehat{\mathbb{E}}_{q_{k}} \left[\Gamma(x) \right] - \mathbb{E}_{\theta_{k}} \left[\Gamma(x) \right].$$

2.2 Extension of GASS to Optimization of CVaR

When CVaR of the loss function $l(x, \xi_x)$, $C_{\alpha^*}(x)$, could be evaluated exactly for all $x \in \mathcal{X}$, we can directly apply the scheme described above to solve the CVaR minimization problem (2.1). Since $l(x, \xi_x)$ is usually evaluated via simulation, generally its p.d.f. and c.d.f. are not available; thus, $C_{\alpha^*}(x)$ could not be evaluated analytically. Nevertheless, it could be estimated via Monte Carlo simulation. In particular, suppose M i.i.d. loss samples $\{l(x, \xi_x^1), l(x, \xi_x^2), ..., l(x, \xi_x^M)\}$ are simulated, and then sorted in ascending order as $l(x, \xi_x^{(1)}) \leq$ $l(x, \xi_x^{(2)}) \leq ... \leq l(x, \xi_x^{(M)})$, which forms an empirical loss distribution. A natural estimator of $C_{\alpha^*}(x)$ is CVaR of the empirical loss distribution, which is defined as follows.

$$\widehat{C}_{\alpha^*}(x) \stackrel{\Delta}{=} \widehat{V}_{\alpha^*}(x) + \frac{1}{M(1-\alpha^*)} \sum_{m=1}^M \left(l(x,\xi_x^m) - \widehat{V}_{\alpha^*}(x) \right)^+, \qquad (2.10)$$

where

$$\widehat{V}_{\alpha^*}(x) \stackrel{\triangle}{=} l\left(x, \xi_x^{(\lceil \alpha^* M \rceil)}\right) \tag{2.11}$$

is VaR of the empirical loss distribution 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 $\widehat{C}_{\alpha^*}(x)$ is biased, it is strongly consistent and asymptotic normally distributed under mild regularity assumptions (see, e.g., Zhu and Zhou (2016)). In principle, we can use it as a replacement for $C_{\alpha^*}(x)$ and plug it into GASS algorithm.

3 Algorithms: GASS-CVaR, GASS-CVaR-ARL

Now let us formally present the following Algorithm 1, which is referred to as GASS-CVaR, for simulation optimization of CVaR.

In the initialization step (step 1) of GASS-CVaR, the conditions on the sample size and step size sequences are imposed to facilitate the convergence of the algorithm. They are typical requirements for a stochastic approximation algorithm. Since in the sampling step (step 2) the CVaR values are estimated, the convergence of the original GASS algorithm, which is designed for deterministic optimization, does not directly apply to GASS-CVaR. We will show the convergence of GASS-CVaR later. In the estimation step (step 3), as mentioned before, one common choice of the shape function $S_{\theta}(\cdot)$ is the one in (2.2). Moreover, the quantile level ρ in (2.3) controls the percentile of elite samples that are used to update the sampling distribution at the next iteration, and balances between the exploitation of the neighborhood of current best solutions and the exploration of the entire solution space. For example, when a smaller ρ is used, less elite samples are used in the updating of the sampling distribution, and thus less emphasis is put on exploration. In the updating step (step 4), the iterate is projected onto a convex and compact subset $\Theta \subseteq \Theta$ 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 Levels

When the risk level of interest α^* is close to 1, implementing GASS-CVaR could be computationally expensive, since in step 2 the CVaR evaluation requires a large sample size M_k

Algorithm 1 Gradient-based Adaptive Stochastic Search 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 $\{M_k\}$ that satisfies $M_k \to \infty$, 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 candidate solutions $\{x_k^i \stackrel{i.i.d.}{\sim} f(x;\theta_k) : i = 1, 2, ..., N_k\}$. For each x_k^i , simulate i.i.d. loss scenarios $\{l(x_k^i, \xi_k^{i,j}) : j = 1, ..., M_k\}$, and sort them in ascending order, denoted by

$$l\left(x_{k}^{i},\xi_{k}^{i,(1)}\right) \leq l\left(x_{k}^{i},\xi_{k}^{i,(2)}\right) \leq \cdots \leq l\left(x_{k}^{i},\xi_{k}^{i,(M_{k})}\right).$$

Estimate the CVaR of the loss for each candidate solution at target risk level α^* :

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

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

$$\widehat{w}_{k}^{i} = \frac{\widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha^{*}}(x_{k}^{i})\right)}{\sum_{j=1}^{N_{k}}\widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha^{*}}(x_{k}^{j})\right)}, \quad i = 1, ..., N_{k}$$

and then estimate $\mathbb{E}_{q_k}[\Gamma(x)]$ and $Var_{\theta_k}[\Gamma(x)]$ via

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

Estimate the gradient g_k by $\widehat{g}_k := \widehat{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)]$. 4. **Updating**: Update the sampling distribution parameter θ according to

Opdating: Opdate the sampling distribution parameter *b* according to

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \widehat{g}_k \right\},\,$$

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. to obtain a good CVaR estimator. This issue is more severe as α^* gets closer to 1. For example, for a fixed x, suppose we want to estimate $C_{\alpha}(x)$ at three different risk levels: $\alpha_1 = 0, \alpha_2 = 0.90$, and $\alpha_3 = 0.99$, where we note that $C_{\alpha_1=0}(x) = \mathbb{E}[l(x,\xi_x)]$ is the expected loss. Loosely speaking, to achieve the same level of accuracy in CVaR estimation, the corresponding sample sizes M_1, M_2 , and M_3 should result in equal "effective" sample sizes. In particular, theoretically using M_i i.i.d. samples to estimate $C_{\alpha_i}(x)$ results in $(1 - \alpha_i)M_i$ effective samples, since the rest α_iM_i samples result in a value of zero. This implies that $(1 - \alpha_1)M_1 = (1 - \alpha_2)M_2 = (1 - \alpha_3)M_3$ for equal effective sample sizes. Thus, $M_2 = (1 - \alpha_1)/(1 - \alpha_2) \cdot M_1 = 10 \cdot M_1$ and $M_3 = (1 - \alpha_1)/(1 - \alpha_3) \cdot M_1 = 100 \cdot M_1$. Therefore, the sample size required for accurate CVaR estimation could be easily up to tens of times even hundreds of times compared with the sample size required for accurate 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 the risk level α_k at every iteration until the target risk level α^* is achieved while the algorithm converges at the same time. Since a lower risk level implies that a smaller M_k is required to achieve the 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 the 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 puts more emphasis on the exploitation of the promising region that has been identified. The key to such an updating rule lies in finding an empirical signal on the algorithm's emphasis between exploration and exploitation.

Note that GASS-CVaR maintains the structure of a gradient-based optimization scheme, and thus the gradient g_k (even the Hessian) used in the updating rule of sampling distribution parameter could be regarded as an empirical signal on the algorithm's balance between exploration and exploitation. Loosely speaking, when the norm of g_k is relatively large, the sampling distribution parameter at next iteration, θ_{k+1} , will differ from θ_k significantly. This means the algorithm is in the "warm-up" phase, where different regions of the solution space are being explored. When the norm of g_k is small, θ_{k+1} is expected to be close to θ_k . This means the algorithm is in the "convergence" phase, where an identified promising region is being exploited. Therefore, it is natural to design the updating rule on risk level using the information contained in the gradient g_k obtained at every iteration. For example, note that GASS-CVaR converges when the norm of g_k hits zero. Then naturally one could increase the risk level at every iteration proportionally to the decrease in the norm of g_k from previous iteration, which ensures that the target risk level α^* is achieved when the gradient hits zero, i.e., when the algorithm converges.

In particular, we propose an updating scheme on the risk level as follows.

$$\alpha_{k+1} = \begin{cases} \alpha^* - \frac{\|g_k\|_2}{\|g_{k-1}\|_2} (\alpha^* - \alpha_k), & \text{if } \|g_k\|_2 < \|g_{k-1}\|_2, \\ \alpha_k, & o/w, \end{cases}$$
(3.1)

where $\|\cdot\|_2$ is the vector Euclidean norm. Note that the updating rule (3.1) ensures that α_k is non-decreasing and bounded above by α^* , with the hope that α_k will eventually converge to α^* . Furthermore, when $\|g_k\|_2 < \|g_{k-1}\|_2$, we can rewrite (3.1) as $\frac{\alpha^* - \alpha_{k+1}}{\alpha^* - \alpha_k} = \frac{\|g_k\|_2}{\|g_{k-1}\|_2}$. Loosely speaking, it implies the increase in the risk level for next iteration is proportional to the decrease in the norm of the gradient from previous iteration. It also ensures that α^* is achieved when the norm of the gradient hits zero, i.e., when the algorithm converges. We do point out that more sophisticated updating rules on the risk level could be incorporated in the future.

Now we present the following Algorithm 2, which is referred to as GASS-CVaR-ARL, for simulation optimization of CVaR with adaptive risk levels.

Algorithm 2 GASS-CVaR with Adaptive Risk Levels

1. Initialization: Initialize the algorithm similar to step 1 of GASS-CVaR. Set initial risk level $\alpha_0 = 0$.

2. Sampling: Draw candidate solutions and simulate the loss distribution scenarios same as step 2 of GASS-CVaR. Estimate $C_{\alpha_k}(x_k^i)$, CVaR of the loss at the risk level α_k , by

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

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

$$\overline{w}_{k}^{i} = \frac{\widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha_{k}}(x_{k}^{i})\right)}{\sum_{j=1}^{N_{k}}\widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha_{k}}(x_{k}^{j})\right)}, \quad i = 1, ..., N_{k}$$

and then estimate $\mathbb{E}_{q_k}[\Gamma(x)]$ and $Var_{\theta_k}[\Gamma(x)]$ via

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

Estimate the gradient g_k by $\overline{g}_k := \overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)]$.

4. Updating: Update the sampling distribution parameter θ according to

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \overline{g}_k \right\},\,$$

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

$$\alpha_{k+1} = \begin{cases} \alpha^* - \frac{\|\bar{g}_k\|_2}{\|\bar{g}_{k-1}\|_2} (\alpha^* - \alpha_k), & \text{if } \|\bar{g}_k\|_2 < \|\bar{g}_{k-1}\|_2, \\ \alpha_k, & o/w. \end{cases}$$
(3.2)

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

In the sampling step (step 2) of GASS-CVaR-ARL, since the current risk level α_k is smaller than the target risk level α^* , we could use a sample size M_k smaller than the one used in GASS-CVaR to estimate the CVaR values at risk level α_k . For example, suppose one wants to keep the "effective" sample size $(1 - \alpha_k)M_k$ 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 best candidate solution generated at each iteration is also recorded, where note that at k^{th} iteration it is identified by the minimum CVaR value at risk level α_k . So it is a good solution to the CVaR minimization problem as if the target risk level is α_k .

In the estimation step (step 3), note that the estimation of $\{w\}$, $\mathbb{E}_{q_k}[\Gamma(x)]$, and thus g_k differs from the estimation in GASS-CVaR, since α_k instead of α^* is used as the risk level at the k^{th} iteration. One could also view \overline{g}_k as an approximation of the gradient for the reformulated problem of $\max_{x \in \mathcal{X}} - C_{\alpha_k}(x)$. This implies that at each step GASS-CVaR-ARL solves a CVaR optimization problem that is structurally similar to the original one but less computationally intensive.

In the updating step (step 4), the updating rule (3.2) is an implementable version of (3.1), with the gradient g_k being replaced by \bar{g}_k . Note that it still ensures that α_k is non-decreasing bounded above by α^* , and α^* is achieved when the norm of \bar{g}_k 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 α^* for all the best candidate solutions found so far, and thus additional simulation budget is required; however, it is insignificant compared with the overall budget consumed.

Recall that, in GASS-CVaR-ARL, the risk level used at each iteration is updated in accordance with the decrease in the norm of the gradient. It implies that the updating rule (3.1) keeps track of the algorithm's balance between the exploration of the entire solution space and the exploitation of an identified promising region, and then makes adjustments on the risk level accordingly. Therefore, in the "warm-up" phase of the algorithm, using a small risk level has little negative effect on the algorithm progress since the algorithm puts most of its emphasis on exploration; in the "convergence" phase of the algorithm, the risk level α_k is close to α^* , and essentially the algorithm is solving problems that are very close to the original one. Thus, intuitively, we expect the number of iterations that GASS-CVaR-ARL takes to converge to be similar to the one that GASS-CVaR takes to converge, which is also verified by the numerical tests presented in Section 5. Since GASS-CVaR-ARL saves simulation budget at every iteration, total budget saving is achieved.

4 Convergence Analysis

Let us first analyze the convergence properties of GASS-CVaR (Algorithm 1). The analysis will rely mainly on the convergence analysis of GASS in Zhou and Hu (2014) as well as the classic results in stochastic approximation methods and algorithms (see, e.g., Kushner and Yin (2003), Borkar (2008), Kushner (2010), and Kushner and Clark (2012)). The main idea is to reformulate the updating scheme on θ_k in GASS-CVaR as a generalized Robbins-Monro recursive algorithm in solving a constrained ordinary differential equation (ODE) of θ , and then show the corresponding bias term and noise term in the reformulated updating scheme are bounded in appropriate asymptotical sense so that the sequence $\{\theta_k\}$ generated by the updating scheme converges to a limit set of the ODE w.p.1.

Following the above road map, let us first reformulate the parameter updating scheme in

GASS-CVaR

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \widehat{g}_k \right\}$$

$$= \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \left(\widehat{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)] \right) \right\},$$

$$(4.1)$$

as

$$\theta_{k+1} = \theta_k + \beta_k \left[G(\theta_k) + b_k + e_k + p_k \right].$$
(4.2)

Here

$$\begin{array}{rcl}
G(\theta_k) & \stackrel{\Delta}{=} & V_k^{-1} \left(\mathbb{E}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)] \right), \\
& b_k & \stackrel{\Delta}{=} & \widehat{V}_k^{-1} \left(\widehat{\mathbb{E}}_{q_k}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] \right), \\
& e_k & \stackrel{\Delta}{=} & \left(\widehat{V}_k^{-1} - V_k^{-1} \right) \left(\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)] \right) + V_k^{-1} \left(\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{q_k}[\Gamma(x)] \right), \\
\end{array}$$

and p_k is the resulted projection error term, where for simplicity we denote

$$V_k \stackrel{ riangle}{=} (Var_{\theta_k}[\Gamma(x)] + \epsilon I) \quad \text{and} \quad \widehat{V}_k \stackrel{ riangle}{=} \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I\right).$$

In (4.2) the term $G(\theta_k)$ is the gradient vector field in a standard stochastic approximation algorithm, the term b_k represents the bias in estimating $\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)]$ caused by the inexact evaluation of the shape function, the term e_k represents the simulation noise in the estimators $\widehat{Var}_{\theta_k}[\Gamma(x)]$ and $\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)]$, and the term p_k represents the projection error after taking the current iterate back onto the constraint set $\widetilde{\Theta}$ with minimum Euclidean norm. Note that the bias term b_k is caused by both the outer-layer sampling on the solution space and the inner-layer simulation of the loss distribution; however, the noise term e_k accounts for the error due to the outer-layer sampling only, since both $\widehat{Var}_{\theta_k}[\Gamma(x)]$ and

$$\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] = \sum_{i=1}^{N_k} w_k^i \Gamma(x_k^i), \quad \text{where} \quad w_k^i = \frac{S_{\theta_k}(-C_{\alpha^*}(x_k^i))}{\sum_{j=1}^{N_k} S_{\theta_k}(-C_{\alpha^*}(x_k^i))}$$

do not involve the inner-layer sampling of $l(x; \xi_x)$.

Now let us introduce the assumptions on the algorithm and $l(x; \xi_x)$ for the convergence of the algorithm. The following set of assumptions is on the algorithm parameters and the choice of the exponential family of densities. It largely follows from the standard assumptions for a generalized stochastic approximation algorithm.

Assumption 1.

- (i) The step size sequence $\{\beta_k\}$ satisfies that $\beta_k > 0$ for all $k, \beta_k \searrow 0$ as $k \to \infty$, $\sum_{k=0}^{\infty} \beta_k = \infty$ and $\sum_{k=0}^{\infty} \beta_k^2 < \infty$.
- (ii) The outer-layer sample size sequence $\{N_k\}$ satisfies $N_k = N_0 \cdot k^{\tau}$ for some constant $\tau > 0$. Furthermore, the sequences $\{\beta_k\}$ and $\{N_k\}$ jointly satisfies $\frac{\beta_k}{\sqrt{N_k}} = O(k^{-\zeta})$ for some constant $\zeta > 1$.
- (iii) The inner-layer sample size sequence M_k satisfies that $M_k \nearrow \infty$ as $k \rightarrow \infty$.

(iv) The sufficient statistics $\Gamma(x)$ of the chosen exponential family of densities is bounded on \mathcal{X} .

In the above set of assumptions, Assumption 1.(i) follows from the typical step size assumption in a gradient-based optimization algorithm. Assumption 1.(ii) ensures that the outer-layer sample size N_k increases to infinity no slower than certain speed given a choice of the step size sequence, and it can be easily satisfied. For example, if $\beta_k = O(1/k)$, then $N_k = N_0 \cdot k^{\tau}$ for an arbitrary constant $\tau > 0$ is sufficient for $\frac{\beta_k}{\sqrt{N_k}} = O(k^{-\zeta})$ to hold for some constant $\zeta > 1$. Assumption 1.(iii) ensures that the error of the CVaR estimators caused by the inner-layer simulation of the loss distribution vanishes as $k \to \infty$. Assumption 1.(iv) is to bound the expectation and variance terms of the sufficient statistics in the algorithm. It holds for many exponential families used in practice. For example, when the solution space \mathcal{X} is a nonempty compact set, the continuity of the function $\Gamma(\cdot)$ will be sufficient for Assumption 1.(iv) to hold.

The next set of assumptions is on the regularity conditions of the loss function $l(x, \xi_x)$. As noted previously, the bias term b_k is caused by the inexact evaluation of the shape function $S_{\theta_k}(\cdot)$. When $S_{\theta_k}(\cdot)$ takes the form of (2.2), b_k is caused by the error in estimating the $(1 - \rho)$ -quantile γ_{θ_k} in (2.3) as well as the error in Monte Carlo estimation of the CVaR values. Specifically, recall that for a fixed x,

$$\widehat{S}_{\theta_k}\left(-\widehat{C}_{\alpha^*}(x)\right) = \frac{1}{1 + \exp\left(-S_o\left(-\widehat{C}_{\alpha^*}(x) - \widehat{\gamma}_{\theta_k}\right)\right)},\tag{4.3}$$

where $\widehat{C}_{\alpha^*}(x)$ is the CVaR estimator given in (2.10), and $\widehat{\gamma}_{\theta_k}$ is the sample $(1-\rho)$ -quantile of $\{-\widehat{C}_{\alpha^*}(x_k^i) : i = 1, ..., N_k\}$, i.e., $\widehat{\gamma}_{\theta_k}$ is the $(\lceil (1-\rho)N_k \rceil)^{th}$ order statistic of $\{-\widehat{C}_{\alpha^*}(x_k^i) : i = 1, ..., N_k\}$. Since γ_{θ_k} could be viewed as the $(1-\rho)$ -level Value-at-Risk (VaR) of $-C_{\alpha^*}(x)$ w.r.t. the sampling distribution $f(x; \theta_k)$, then $\widehat{\gamma}_{\theta_k}$ could be regarded as a nested risk estimator in which the outer-layer simulation is on estimation of VaR and the inner-layer is on estimation of CVaR. Hence, bounding the bias term b_k reduces to bounding the errors of the nested risk estimator $\widehat{\gamma}_{\theta_k}$ as well as the one-layer CVaR estimator $\widehat{C}_{\alpha^*}(x)$. Here we will resort to the asymptotic analysis of nested risk estimators in Gordy and Juneja (2010), and Zhu and Zhou (2016).

To this end, let us rewrite the CVaR estimator $\widehat{C}_{\alpha^*}(x)$ in (2.10) as

$$\widehat{C}_{\alpha^*}(x) = C_{\alpha^*}(x) + \frac{1}{\sqrt{M_k}} \cdot \mathcal{E}_k(x), \quad \forall x \in \mathcal{X},$$

where $\mathcal{E}_k(x)$ is the standardized error of the CVaR estimator. Note that by the asymptotic normality of $\widehat{C}_{\alpha^*}(x)$, under appropriate regularity conditions $\mathcal{E}_k(x)$ has a limiting distribution as $k \to \infty$. Thus, the effect of the diminishing noise term $\mathcal{E}_k(x)/\sqrt{M_k}$ on the distribution of $\widehat{C}_{\alpha^*}(x)$ will vanish as $M \to \infty$. Hence, we expect the "distance" between the distribution of $\widehat{C}_{\alpha^*}(x)$ and the distribution of $C_{\alpha^*}(x)$ to vanish as $M \to \infty$. That is, the p.d.f. of $\widehat{C}_{\alpha^*}(x)$ converges to the p.d.f. of $C_{\alpha^*}(x)$. The following set of assumptions, which is referred to Assumption 2, guarantees that the convergence of the p.d.f. is sufficiently fast. It largely follows from Assumption 1 in Gordy and Juneja (2010) and Assumption 3.2 in Zhu and Zhou (2016).

Assumption 2.

- (i) For all $x \in \mathcal{X}$, the loss distribution $l(x, \xi_x)$ has finite second moment; moreover, for all $\theta \in \Theta$, the CVaR function $C_{\alpha^*}(x)$, which is a random variable under the distribution $f(\cdot; \theta)$, has finite second moment.
- (ii) For all $\theta \in \Theta$ and each k, the joint density $d_k(c, e)$ of $(C_{\alpha^*}(x), \mathcal{E}_k(x))$, and its partial derivatives $\frac{\partial}{\partial c} d_k(c, e)$ and $\frac{\partial^2}{\partial c^2} d_k(c, e)$ exist for all pairs of (c, e).
- (iii) For all $\theta \in \Theta$ and each k, there exist nonnegative functions $D_{0,k}(\cdot)$, $D_{1,k}(\cdot)$ and $D_{2,k}(\cdot)$ such that $d_k(c, e) < D_{0,k}(e)$, $\frac{\partial}{\partial c}d_k(c, e) < D_{1,k}(e)$, and $\frac{\partial^2}{\partial c^2}d_k(c, e) < D_{2,k}(e)$ for all (c, e). Furthermore, for all $\theta \in \Theta$, $\sup_k \int |e|^r D_{i,k}(e) de < \infty$ for i = 0, 1, 2, and $0 \le r \le 4$.

In the above assumption, Assumption 2.(i) ensures that a one-layer VaR or CVaR estimator defined in (2.10) or (2.11) is strongly consistent and asymptotically normally distributed, and thus the standardized estimation error $\mathcal{E}_k(x)$ has a limiting distribution as $k \to \infty$. Assumption 2.(ii) and 2.(iii) further ensure that the p.d.f. of $\widehat{C}_{\alpha^*}(x)$ converges to the p.d.f. of $C_{\alpha^*}(x)$ sufficiently fast. This will imply the strong consistency of the nested risk estimator $\widehat{\gamma}_{\theta_k}$ and further the convergence of the approximate shape function $\widehat{S}_{\theta_k}\left(-\widehat{C}_{\alpha^*}(x)\right)$, as presented in the following Lemma 4.1.

Lemma 4.1. Suppose the shape function $S_{\theta_k}(\cdot)$ takes the form

$$S_{\theta_{k}}(-C_{\alpha^{*}}(x)) = \frac{1}{1 + \exp(-S_{o}(-C_{\alpha^{*}}(x) - \gamma_{\theta_{k}}))}$$

where S_o is a large positive constant, and $\gamma_{\theta_k} \stackrel{\Delta}{=} \sup_r \left\{ r : P_{f(\cdot;\theta_k)} \left\{ x \in \mathcal{X} : -C_{\alpha^*}(x) \geq r \right\} \geq \rho \right\}$ is the $(1 - \rho)$ -quantile of $(-C_{\alpha^*}(x))$ w.r.t. $f(\cdot;\theta_k)$. Further suppose that $S_{\theta_k}(-C_{\alpha^*}(x))$ is approximated by $\widehat{S}_{\theta_k}\left(-\widehat{C}_{\alpha^*}(x)\right)$ as in (4.3). Then under Assumption 1.(ii), 1.(iii) and Assumption 2, we have

$$\lim_{k \to \infty} \left| \widehat{S}_{\theta_k} \left(-\widehat{C}_{\alpha^*}(x) \right) - S_{\theta_k} \left(-C_{\alpha^*}(x) \right) \right| = 0, \quad w.p.1, \quad \forall x \in \mathcal{X}.$$

$$(4.4)$$

The main idea of the proof is to show $\widehat{C}_{\alpha^*}(x) \to C_{\alpha^*}(x)$ w.p.1 and $\widehat{\gamma}_{\theta_k} \to \gamma_{\theta_k}$ w.p.1 as $k \to \infty$. The detailed proof is included in the appendix. Following the road map and based on Lemma 1, we next show that the bias term b_k converges to zero w.p.1. as $k \to \infty$, as presented in Lemma 4.2 below.

Lemma 4.2. Under Assumption 1 and Assumption 2, we have

$$\lim_{k \to \infty} \|b_k\|_2 = 0, \ w.p.1, \tag{4.5}$$

where recall that $||b_k||_2$ is the vector Euclidean norm of b_k .

The proof of Lemma 4.2 is included in the appendix. Continuing the road map, we next show that the summed tail error goes to zero w.p.1, as presented in the following Lemma 4.3.

Lemma 4.3. Under Assumption 1, we have

$$\lim_{k \to \infty} \left\{ \sup_{\substack{n: 0 \le \sum_{i=k}^{n-1} \beta_i \le T}} \left\| \sum_{i=k}^n \beta_i e_i \right\|_2 \right\} = 0, \quad w.p.1$$

$$(4.6)$$

for all $T \geq 0$.

Lemma 4.3 is identical to Lemma 2 in Zhou and Hu (2014), so we omit the proof here. With the above lemmas, we now proceed to the main result on the convergence of Algorithm 1.

Given an arbitrary $\theta \in \widetilde{\Theta}$, a set $\mathcal{C}(\theta)$ is defined as follows. For θ that lies in the interior of $\widetilde{\Theta}$, let $\mathcal{C}(\theta) = \{0\}$; for θ that lies on the boundary of $\widetilde{\Theta}$, let $\mathcal{C}(\theta)$ be the infinite convex cone generated by the outer normals at θ of the faces on which θ lies (see, e.g., Kushner (2010) pp. 89). Then the updating scheme (4.2) in GASS-CVaR could be viewed as a noisy discretization of a constrained ODE for $\{\theta(t) : t \geq 0\}$:

$$\dot{\theta}(t) = G(\theta(t)) + p(t), \ p(t) \in -\mathcal{C}(\theta(t)), \ t \ge 0,$$

$$(4.7)$$

where p(t) is the minimum force to take $\theta(t)$ back to the set Θ . Using the ODE approach for the convergence of the Robbins-Monro Algorithm (see, e.g., Kushner (2010)), we can show that the sequence $\{\theta_k\}$ generated by (4.1) converges to a limit set of the ODE (4.7). In particular, we have the following theorem.

Theorem 4.1. Convergence of GASS-CVaR. Suppose Assumption 1 and Assumption 2 hold. Then the sequence $\{\theta_k\}$ generated by (4.1) converges to a limit set of the ODE (4.7) w.p.1. Furthermore, if the limit sets of (4.7) are isolated equilibrium points, then $\{\theta_k\}$ converges to a unique equilibrium point w.p.1.

Theorem 4.1 is a direct consequence of Theorem 2 in Kushner (2010) with Lemma 4.2 and Lemma 4.3 above. Starting with the convergence of GASS-CVaR, we will show the convergence of the algorithm GASS-CVaR-ARL. The intuition is as follows.

Recall that the updating scheme on θ in GASS-CVaR-ARL is

$$\theta_{k+1} = \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \overline{g}_k \right\}$$

$$= \Pi_{\widetilde{\Theta}} \left\{ \theta_k + \beta_k \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I \right)^{-1} \left(\overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \mathbb{E}_{\theta_k}[\Gamma(x)] \right) \right\}.$$

$$(4.8)$$

Compared with the updating scheme (4.1) on θ in GASS-CVaR, we could see that the approximate expectation term $\widehat{\mathbb{E}}_{q_k}[\Gamma(x)]$ in (4.1) is replaced by $\overline{\mathbb{E}}_{q_k}[\Gamma(x)]$ in (4.8) in estimating the gradient g_k . Note that the updating scheme for the risk level α_k in (3.2) guarantees that α_k is non-decreasing and bounded above by the target risk level α^* . Thus, the limit of the risk level sequence $\{\alpha_k\}$ exists. If we are able to show that the limit is α^* , then the difference between $\overline{\mathbb{E}}_{q_k}[\Gamma(x)]$ and $\widehat{\mathbb{E}}_{q_k}[\Gamma(x)]$, i.e., the difference between \overline{g}_k and \widehat{g}_k , will vanish as $k \to \infty$. The reason is that the normalized weights $\{\overline{w}_k\}$ in computing $\overline{\mathbb{E}}_{q_k}[\Gamma(x)]$ will asymptotically approach $\{\widehat{w}_k\}$ in computing $\widehat{\mathbb{E}}_{q_k}[\Gamma(x)]$ as $k \to \infty$.

Assume by contradiction that $\lim_{k\to\infty} \alpha_k = \bar{\alpha}^* < \alpha^*$. On the one hand, following from above argument, GASS-CVaR-ARL asymptotically approaches GASS-CVaR for the simulation optimization of $C_{\bar{\alpha}^*}(x)$ instead of $C_{\alpha^*}(x)$. Therefore, it is convergent, and thus the gradient sequence $\{g_k\}$ approaches zero w.p.1. One the other hand, the sequence $\{\|\bar{g}_k\|_2\}$ generated by (3.2) will always be above a certain positive value w.p.1 (otherwise α_k will converge to α^*). This contradicts with the fact that $\{g_k\}$ approaches zero w.p.1. We formalize the above analysis in the following Theorem 4.2. The detailed proof is included in the appendix. **Theorem 4.2.** Convergence of GASS-CVaR-ARL. Suppose Assumption 1 and Assumption 2 hold. Then the risk level sequence $\{\alpha_k\}$ generated by (3.2) converges to the target risk level α^* w.p.1, and the sequence $\{\theta_k\}$ generated by (4.8) converges to a limit set of the ODE (4.7) w.p.1. Furthermore, if the limit sets of (4.7) are isolated equilibrium points, then $\{\theta_k\}$ converges to a unique equilibrium point w.p.1.

5 Numerical Experiments

We carry out numerical tests to compare the performances 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 or 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

$$l_i(x,\xi_x) = L_i(x) + \begin{cases} \sqrt{1+100\sum_{d=1}^{D}(x_d-1)^2} \cdot \mathcal{N}(0,1), \ i = 0, 1, 3, 4, \\ \sqrt{1+100\sum_{d=1}^{D}(x_d-2)^2} \cdot \mathcal{N}(0,1), \ i = 2, 5, \end{cases}$$
(5.1)

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. The explicit expressions of L_i 's are listed as follows, and we test all functions with D = 10.

- (0) $L_0(x) = \sum_{d=1}^{D} x_d^2$.
- (1) Powell function $L_1(x)$.

$$L_1(x) = \sum_{d=2}^{D-2} \left[(x_{d-1} + 10x_d)^2 + 5(x_{d+1} - x_{d+2})^2 + (x_d - 2x_{d+1})^4 + 10(x_{d-1} - x_{d+2})^4 \right].$$

(2) Rosenbrock function $L_2(x)$.

$$L_2(x) = \sum_{d=1}^{D-1} \left[(x_d - 1)^2 + 100(x_d^2 - x_{d+1})^2 \right].$$

(3) Rastrigin function $L_3(x)$.

$$L_3(x) = \sum_{d=1}^{D} (x_d^2 - 10\cos(2\pi x_d)) - 10D - 1.$$

(4) Pintér function $L_4(x)$.

$$L_4(x) = \left[\sum_{d=1}^{D} dx_d^2 + \sum_{d=1}^{D} 20d\sin^2(x_{d-1}\sin x_d - x_d + \sin x_{d+1})\right]$$

+
$$\sum_{d=1}^{D} d \log_{10} (1 + d(x_{d-1}^2 - 2x_d + 3x_{d+1} - \cos x_d + 1)^2)].$$

(5) Levy function $L_5(x)$.

$$L_5(x) = -\sin^2(\pi y_1) - \sum_{d=1}^{D-1} \left[(y_d - 1)^2 (1 + 10\sin^2(\pi y_d + 1)) \right] - (y_D - 1)^2 (1 + 10\sin^2(2\pi y_D)),$$

where $y_d = 1 + (x_d - 1)/4$, d = 1, ..., D.

Note that we add the noise in the above form to make sure the optimal solution becomes different when α^* varies. If $C_{\alpha^*=0}(x) = L_i(x) = \mathbb{E}_{\xi_x}[l_i(x,\xi_x)]$ 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 $C_{\alpha^*}(x)$, might be very different from x° . Specifically, the loss distribution of l_i has a relatively large variance at x° (note that it has the smallest variance at $x = [x_1^o + 1, ..., x_D^o + 1]$). This indicates that, as the risk level of interest α^* increases, the minimizer of $C_{\alpha^*}(x)$ may start to deviate away from x° and move towards $x = [x_1^o + 1, ..., x_D^o + 1]$ (this is also verified by our numerical tests), where the loss function is exposed to the lowest amount of noise. Note that when $\alpha^* > 0$, except for l_0 , the minimizers of $C_{\alpha^*}(x)$ and the minimum CVaR function values are not analytically available.

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 $\Sigma_0 = 1000I_{D\times D}$, 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 (2.2) with $S_o = 10^5$ and $\rho = 0.1$ in (2.3). 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 sample size used to estimate the CVaR of the loss distribution is set in a way such that the effective sample size 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 sample size $M_0 = 50/(1 - 0) = 50$. The small positive constant ϵ used to ensure the positive definiteness of the Hessian is set to be $\epsilon = 10^{-10}$, 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 large sample sizes $N = 5 \times 10^3$ and $M = 10^5$ to find close approximations of the true minimum CVaR values, which will be served as benchmark values later when comparing algorithm performance. Later the approximate minimum CVaR values will be used as the true CVaR values in comparing the algorithm performance. In the upper-left plot of Figure 1 for the loss function l_0 , the y-axis represents the ratio of the best CVaR values found by the algorithms to the 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 minimum is replaced by the approximate minimum CVaR values from implementing GASS-CVaR with sample sizes N and 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 levels 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 minimum CVaR value; 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 $\alpha^* = 0.99$, which indicates that the norm of the gradient decreases steadily to zero and the algorithm converges.

6 Conclusion

In this paper, we study the extension of the recently proposed algorithm GASS, which is designed for the optimization of deterministic non-differentiable objectives, to the simulation optimization of risk measures such as VaR and CVaR. Instead of optimizing VaR or 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 such that the target risk level is achieved while the algorithm converges simultaneously. It enables us to adaptively reduce the number of samples needed to estimate VaR or CVaR at each iteration, leading to improvement of efficiency over the original algorithm. The numerical results demonstrate the advantage of incorporating such an adaptive updating rule on the risk level in the algorithm by showing it results in a 2-4 times of total budget saving for the tested loss functions.

Acknowledgements

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.

A Proof of Theorems

Proof. **Proof of Lemma 4.1.** Since $S_{\theta_k}(\cdot)$ is continuous in both C_{α^*} and γ_{θ_k} , it suffices to show that for all $x \in \mathcal{X}$

$$\lim_{k \to \infty} \widehat{C}_{\alpha^*}(x) \to C_{\alpha^*}(x), \ w.p.1., \quad \text{and} \quad \lim_{k \to \infty} \widehat{\gamma}_{\theta_k} \to \gamma_{\theta_k}, \ w.p.1.$$
(A.1)



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

Let us first show the left part of the above statement. Recall that by Assumption 1.(iii), we have $M_k \to \infty$ as $k \to \infty$. Then, we only need to show that the one-layer CVaR estimator $\hat{C}_{\alpha^*}(x)$ is strongly consistent. By Lemma A.1 in Zhu and Zhou (2016) this holds, where note that Assumption 3.1 in Zhu and Zhou (2016) is satisfied by Assumption 2 here.

It remains to establish the right part of (A.1). In view of Assumption 1.(ii) and 1.(iii), we have $N_k, M_k \to \infty$ as $k \to \infty$. That is, N_k, M_k go to infinity simultaneously as $k \to \infty$. Therefore, it suffices to show

$$\lim_{N_k, M_k \to \infty} \widehat{\gamma}_{\theta_k} \to \gamma_{\theta_k}, \ w.p.1.$$
(A.2)



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

Note that

$$\gamma_{\theta_k} = V_{1-\rho}(-C_{\alpha^*}(x)),$$

i.e., the $(1 - \rho)$ -level Value-at-Risk (VaR) of $(-C_{\alpha^*}(x))$ w.r.t. $f(x; \theta_k)$. Furthermore,

$$\widehat{\gamma}_{\theta_k} = \widehat{V}_{1-\rho}(-\widehat{C}_{\alpha^*}(x)),$$

i.e., the sample $(1 - \rho)$ -quantile of $\{-\widehat{C}_{\alpha^*}(x_k^i) : i = 1, ..., N_k\}$. Therefore, $\widehat{\gamma}_{\theta_k}$ is a nested estimator of γ_{θ_k} , where N_k outer-layer samples are drawn, and for each outer-layer sample M_k inner-layer samples are drawn.

Rewrite $\widehat{C}_{\alpha^*}(x)$ as

$$\widehat{C}_{\alpha^*}(x) = C_{\alpha^*}(x) + \frac{1}{\sqrt{M_k}} \cdot \mathcal{E}_k(x), \quad \forall x \in \mathcal{X},$$
(A.3)

where $\mathcal{E}_k(x)$ is the standardized error. By Theorem 3.3 in Zhu and Zhou (2016), we have

$$\lim_{M_k \to \infty} \sqrt{M_k} \left(\widehat{C}_{\alpha^*}(x) - C_{\alpha^*}(x) \right) \stackrel{\mathcal{D}}{\Rightarrow} \mathcal{N} \left(0, \sigma^2(x) \right),$$

where " $\overset{\mathcal{D}}{\Rightarrow}$ " denotes the convergence in distribution, and $\mathcal{N}(0, \sigma^2(x))$ denotes a normal distribution with mean zero and variance $\sigma^2(x)$, where $\sigma^2(x)$ is the variance parameter that only depends on x. Combined with (A.3), we can see that the standardized error $\mathcal{E}_k(x)$ converges to $\mathcal{N}(0, \sigma^2(x))$ in distribution. Have establishing this, the remaining proof is identical to the proof of Theorem 3.2 in Zhu and Zhou (2016), where note that Assumption 2 here is parallel with Assumption 3.2 in Zhu and Zhou (2016).

Proof. **Proof of Lemma 4.2**. With a slight abuse of notation, we also use $||A||_2$ to denote the spectral norm of a real square matrix A induced by the vector Euclidean norm. In particular, $||A||_2 = \sqrt{\lambda_{max}(A^T A)}$, i.e., $||A||_2$ is the largest eigenvalue of the positive-semidefinite matrix $A^T A$. When the matrix A is positive-semidefinite, $||A||_2$ is just the largest eigenvalue of A.

To facilitate the proof, let us also introduce the following notations:

$$\begin{split} \widetilde{\mathbf{Y}}_{k} &\stackrel{\Delta}{=} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} S_{\theta_{k}} \left(-C_{\alpha^{*}}(x_{k}^{i}) \right) \Gamma(x_{k}^{i}), \quad \widetilde{\mathbb{Z}}_{k} \stackrel{\Delta}{=} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} S_{\theta_{k}} \left(-C_{\alpha^{*}}(x_{k}^{i}) \right), \\ \widehat{\mathbf{Y}}_{k} &\stackrel{\Delta}{=} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} \widehat{S}_{\theta_{k}} \left(-\widehat{C}_{\alpha^{*}}(x_{k}^{i}) \right) \Gamma(x_{k}^{i}), \quad \widehat{\mathbb{Z}}_{k} \stackrel{\Delta}{=} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} \widehat{S}_{\theta_{k}} \left(-\widehat{C}_{\alpha^{*}}(x_{k}^{i}) \right) \end{split}$$

Here note that $\widetilde{\mathbb{Y}}_k$, $\widehat{\mathbb{Y}}_k$ are vectors because $\Gamma(\cdot)$ are vector-valued functions, and $\widetilde{\mathbb{Z}}_k$, $\widehat{\mathbb{Z}}_k$ are scalar-valued.

Since $C_{\alpha^*}(x)$ and $\Gamma(x)$ are both bounded on \mathcal{X} , we immediately have $|\widetilde{\mathbb{Z}}_k|$ bounded below from zero and $\frac{\|\widehat{\mathbb{Y}}_k\|_2}{|\widehat{\mathbb{Z}}_k|}$ bounded for all k. Note that

$$b_{k} = \widehat{V}_{k}^{-1} \left(\widehat{\mathbb{E}}_{q_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}[\Gamma(x)] \right)$$

$$= \widehat{V}_{k}^{-1} \left(\frac{\widehat{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} - \frac{\widetilde{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} \right)$$

$$= \widehat{V}_{k}^{-1} \left(\frac{\widehat{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} - \frac{\widehat{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} + \frac{\widehat{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} - \frac{\widetilde{\mathbb{Y}}_{k}}{\widehat{\mathbb{Z}}_{k}} \right)$$

$$= \widehat{V}_{k}^{-1} \widehat{\mathbb{Y}}_{k} \left(\frac{\widetilde{\mathbb{Z}}_{k} - \widehat{\mathbb{Z}}_{k}}{\widehat{\mathbb{Z}}_{k} \widehat{\mathbb{Z}}_{k}} \right) + \widehat{V}_{k}^{-1} \frac{\widehat{\mathbb{Y}}_{k} - \widetilde{\mathbb{Y}}_{k}}{\widetilde{\mathbb{Z}}_{k}}.$$

Therefore,

$$\begin{split} \|b_{k}\|_{2} &\leq \frac{\|\widehat{V}_{k}^{-1}\|_{2}}{|\widetilde{Z}_{k}|} \frac{\|\widehat{Y}_{k}\|_{2}}{|\widehat{Z}_{k}|} \left|\widetilde{Z}_{k} - \widehat{Z}_{k}\right| + \frac{\|\widehat{V}_{k}^{-1}\|_{2}}{|\widetilde{Z}_{k}|} \left|\widehat{Y}_{k} - \widetilde{Y}_{k}\right| \\ &\leq \frac{\|\widehat{V}_{k}^{-1}\|_{2}}{|\widetilde{Z}_{k}|} \frac{\|\widehat{Y}_{k}\|_{2}}{|\widehat{Z}_{k}|} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} \left|S_{\theta_{k}}\left(-C_{\alpha^{*}}(x_{k}^{i})\right) - \widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha^{*}}(x_{k}^{i})\right)\right| \\ &+ \frac{\|\widehat{V}_{k}^{-1}\|_{2}}{|\widetilde{Z}_{k}|} \frac{1}{N_{k}} \sum_{i=1}^{N_{k}} \left|\widehat{S}_{\theta_{k}}\left(-\widehat{C}_{\alpha^{*}}(x_{k}^{i})\right) - S_{\theta_{k}}\left(-C_{\alpha^{*}}(x_{k}^{i})\right)\right| \left\|\Gamma(x_{k}^{i})\right\|_{2}. \end{split}$$

Recall that $\widehat{V}_k = \left(\widehat{Var}_{\theta_k}[\Gamma(x)] + \epsilon I\right)$. Thus, it is a positive-definite matrix and its minimum eigenvalue is at least ϵ . It follows that the maximum eigenvalue of \widehat{V}_k^{-1} is no greater than ϵ^{-1} , i.e., $\left\|\widehat{V}_k^{-1}\right\|_2 \leq \epsilon^{-1}$. Since $|\widetilde{Z}_k|$ is bounded below from zero, $\frac{\|\widehat{Y}_k\|_2}{|\widehat{Z}_k|}$ is bounded, and $\Gamma(x)$ is bounded on \mathcal{X} , Lemma 4.1 implies that $\|b_k\|_2 \to 0$ w.p.1 as $k \to \infty$.

Proof. **Proof of Theorem 4.2**. Let us first show the following lemma.

Lemma A.1. Suppose Assumption 1 and Assumption 2 hold. Further suppose the risk level sequence $\{\alpha_k\}$ generated by (3.2) converges to the target risk level α^* w.p.1. Then the sequence $\{\theta_k\}$ generated by (4.8) converges to a limit set of the ODE (4.7) w.p.1.

Proof of Lemma A.1. Similar to the proof of Theorem 4.1, we will reformulate the updating scheme (4.8) as a noisy discretization of the constrained ODE (4.7), and show both the bias and the noise are properly bounded. Specifically, rewrite (4.8) as

$$\theta_{k+1} = \theta_k + \beta_k \left[G(\theta_k) + \bar{b}_k + e_k + \bar{p}_k \right], \tag{A.4}$$

where $G(\theta_k)$ and e_k are defined as previously, $\bar{b}_k \stackrel{\Delta}{=} \widehat{V}_k^{-1} \left(\overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] \right)$, and \bar{p}_k is the projection error term that takes the current iterate back onto the constraint set Θ with minimum Euclidean norm. In view of Theorem 2 in Kushner (2010), it suffices to show

$$\lim_{k \to \infty} \left\| \bar{b}_k \right\|_2 = 0, \quad w.p.1.$$

To ease the presentation, let us denote

$$\widetilde{\mathbb{E}}_{q_k}^{\alpha}[\Gamma(x)] \stackrel{\triangle}{=} \sum_{i=1}^{N_k} \frac{S_{\theta_k}\left(-C_{\alpha}(x_k^i)\right)}{\sum_{j=1}^{N_k} S_{\theta_k}\left(-C_{\alpha}(x_k^j)\right)} \Gamma(x_k^i).$$

It immediately implies that $\widetilde{\mathbb{E}}_{q_k}[\Gamma(x)] = \widetilde{\mathbb{E}}_{q_k}^{\alpha^*}[\Gamma(x)]$. Furthermore,

$$\begin{aligned} \|\overline{b}_{k}\|_{2} &= \left\|\widehat{V}_{k}^{-1}\left(\overline{\mathbb{E}}_{q_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}[\Gamma(x)]\right)\right\|_{2} \\ &= \left\|\widehat{V}_{k}^{-1}\left(\overline{\mathbb{E}}_{q_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}^{\alpha_{k}}[\Gamma(x)]\right) + \widehat{V}_{k}^{-1}\left(\widetilde{\mathbb{E}}_{q_{k}}^{\alpha_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}^{\alpha^{*}}[\Gamma(x)]\right)\right\|_{2} \\ &\leq \left\|\widehat{V}_{k}^{-1}\left(\overline{\mathbb{E}}_{q_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}^{\alpha_{k}}[\Gamma(x)]\right)\right\|_{2} + \left\|\widehat{V}_{k}^{-1}\right\|_{2}\left\|\widetilde{\mathbb{E}}_{q_{k}}^{\alpha_{k}}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_{k}}^{\alpha^{*}}[\Gamma(x)]\right\|_{2}. (A.5) \end{aligned}$$

Following an argument almost identical to the proof of Lemma 4.2, the first term in (A.5) converges to 0 w.p.1 as $k \to \infty$. Note that $S_{\theta_k}(\cdot)$ is a continuous function and $C_{\alpha_k}(x)$ is continuous in α_k . Thus, $\widetilde{\mathbb{E}}_{q_k}^{\alpha_k}[\Gamma(x)]$ is a continuous function in α_k . Therefore, the second term in (A.5) converges to 0 w.p.1 as $k \to \infty$ since $\|\widehat{V}_k^{-1}\|_2$ is bounded and α_k converges to α^* as $k \to \infty$. Proof of Lemma A.1 is now complete.

In view of Lemma A.1, it remains to show that the risk level sequence $\{\alpha_k\}$ generated by (3.2) converges to the target risk level α^* w.p.1. Proof by contradiction. Since the sequence $\{\alpha_k\}$ is non-decreasing and bounded above by α^* , let us assume $\lim_{k\to\infty} \alpha_k = \overline{\alpha}^*$ and $\overline{\alpha}^* < \alpha^*$ w.p.1. Conditioning on this, Lemma A.1 still holds when the target risk level α^* is replaced by $\overline{\alpha}^*$. That is, the algorithm GASS-CVaR-ARL converges, and the gradient sequence $\{g_k\}$ converges to 0 w.p.1. as $k \to \infty$. Note that g_k is bounded (since $\Gamma(x)$ is bounded), by bounded convergence theorem we have

$$\lim_{k \to \infty} \mathbb{E}\left[\|g_k\|_2 \right] = 0. \tag{A.6}$$

Furthermore, note that

$$\mathbb{E}\left[\left\|\overline{g}_{k}-g_{k}\right\|_{2}\right] = \mathbb{E}\left[\left\|\overline{\mathbb{E}}_{q_{k}}[\Gamma(x)]-\mathbb{E}_{q_{k}}^{\overline{\alpha}^{*}}[\Gamma(x)]\right\|_{2}\right] \\
\leq \mathbb{E}\left[\left\|\overline{\mathbb{E}}_{q_{k}}[\Gamma(x)]-\widetilde{\mathbb{E}}_{q_{k}}^{\overline{\alpha}^{*}}[\Gamma(x)]\right\|_{2}\right] + \mathbb{E}\left[\left\|\widetilde{\mathbb{E}}_{q_{k}}^{\overline{\alpha}^{*}}[\Gamma(x)]-\mathbb{E}_{q_{k}}^{\overline{\alpha}^{*}}[\Gamma(x)]\right\|_{2}\right], (A.7)$$

where

$$\mathbb{E}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] \stackrel{\triangle}{=} \frac{\int S_{\theta_k} \left(-C_{\overline{\alpha}^*}(x) \right) \Gamma(x) f(x;\theta_k) dx}{\int S_{\theta_k} \left(-C_{\overline{\alpha}^*}(x) \right) f(x;\theta_k) dx}$$

We have shown in the proof of Lemma A.1 that

$$\lim_{k \to \infty} \left\| \overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] \right\|_2 = 0, \quad w.p.1.$$

Since $\left\|\overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)]\right\|_2$ is bounded, again by bounded convergence theorem

$$\lim_{k \to \infty} \mathbb{E}\left[\left\| \overline{\mathbb{E}}_{q_k}[\Gamma(x)] - \widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] \right\|_2 \right] = 0.$$
(A.8)

Moreover, notice that $\widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)]$ is a self-normalized importance sampling estimator of $\mathbb{E}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)]$. Applying Theorem 9.1.10 (pp. 294) in Cappé et al. (2005), we have

$$\mathbb{E}\left[\left|\widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma_j(x)] - \mathbb{E}_{q_k}^{\overline{\alpha}^*}[\Gamma_j(x)]\right|^2\right] \le \frac{c_j}{N_k}, \ j = 1, ..., d_{\theta},$$

where $\Gamma_j(x)$ is the j^{th} element in the vector $\Gamma(x)$, and c_j 's are positive constants that depend on the bounds of $\Gamma_j(x)$'s on \mathcal{X} . Therefore, by Cauchy-Schwarz Inequality we have

$$\mathbb{E}\left[\left\|\widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] - \mathbb{E}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)]\right\|_2\right] \le \sqrt{\frac{d \cdot \max_j c_j}{N_k}}$$

That is,

$$\lim_{k \to \infty} \mathbb{E}\left[\left\| \widetilde{\mathbb{E}}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] - \mathbb{E}_{q_k}^{\overline{\alpha}^*}[\Gamma(x)] \right\|_2 \right] = 0.$$
(A.9)

Combining (A.7), (A.8) with (A.9), we have

$$\lim_{k \to \infty} \mathbb{E}\left[\left\| \bar{g}_k - g_k \right\|_2 \right] = 0.$$

In view of (A.6), we have

$$\lim_{k \to \infty} \mathbb{E}\left[\|\bar{g}_k\|_2 \right] = 0. \tag{A.10}$$

Since $\bar{\alpha}^* < \alpha^*$, the sequence $\{\|\bar{g}_k\|_2\}$ generated by (3.2) will always be above a certain positive value w.p.1 (otherwise α_k will converge to α^*), which contradicts with (A.10). Proof is complete.

References

- Alexander, Siddharth, Thomas F Coleman, Yuying Li. 2006. Minimizing cvar and var for a portfolio of derivatives. *Journal of Banking & Finance* **30**(2) 583–605.
- Artzner, Philippe, Freddy Delbaen, Jean-Marc Eber, David Heath. 1999. Coherent measures of risk. Mathematical Finance 9 203–228.
- Borkar, Vivek S. 2008. Stochastic approximation: A dynamical systems viewpoint.
- Cappé, Olivier, Eric Moulines, Tobias Rydén. 2005. Inference in hidden Markov models. Springer Series in Statistics, Springer, New York.
- Dorigo, Marco, Christian Blum. 2005. Ant colony optimization theory: A survey. *Theoretical Computer Science* **344**(2) 243–278.
- Gordy, Michael B, Sandeep Juneja. 2010. Nested simulation in portfolio risk measurement. *Management Science* **56**(10) 1833–1848.

- Hu, Jiaqiao, Michael C Fu, Steven I Marcus. 2007. A model reference adaptive search method for global optimization. Operations Research 55(3) 549–568.
- Hu, Jiaqiao, Michael C Fu, Steven I Marcus, et al. 2008. A model reference adaptive search method for stochastic global optimization. *Communications in Information & Systems* 8(3) 245–276.
- Kushner, Harold. 2010. Stochastic approximation: a survey. Wiley Interdisciplinary Reviews: Computational Statistics 2(1) 87–96.
- Kushner, Harold, G George Yin. 2003. Stochastic approximation and recursive algorithms and applications, vol. 35. Springer Science & Business Media.
- Kushner, Harold Joseph, Dean S Clark. 2012. Stochastic approximation methods for constrained and unconstrained systems, vol. 26. Springer Science & Business Media.
- Larranaga, Pedro, Jose A Lozano. 2002. Estimation of distribution algorithms: A new tool for evolutionary computation, vol. 2. Springer Science & Business Media.
- Molvalioglu, Orcun, Zelda B Zabinsky, Wolf Kohn. 2009. The interacting-particle algorithm with dynamic heating and cooling. *Journal of Global Optimization* **43**(2-3) 329–356.
- Molvalioglu, Orcun, Zelda B Zabinsky, Wolf Kohn. 2010. Meta-control of an interacting-particle algorithm for global optimization. *Nonlinear Analysis: Hybrid Systems* 4(4) 659–671.
- Rockafellar, R Tyrrell, Stanislav Uryasev. 2000. Optimization of conditional value-at-risk. *Journal* of Risk **2** 21–42.
- Rockafellar, R Tyrrell, Stanislav Uryasev. 2002. Conditional value-at-risk for general loss distributions. Journal of Banking & Finance 26(7) 1443–1471.
- Romeijn, H Edwin, Robert L Smith. 1994. Simulated annealing for constrained global optimization. Journal of Global Optimization 5(2) 101–126.
- Rubinstein, Reuven Y. 2001. Combinatorial optimization, cross-entropy, ants and rare events. Stochastic Optimization: Algorithms and Applications 54 303–363.
- Ruszczyński, Andrzej. 2010. Risk-averse dynamic programming for markov decision processes. Mathematical Programming 125(2) 235–261.
- Ruszczyński, Andrzej, Alexander Shapiro. 2006. Optimization of convex risk functions. *Mathematics of Operations Research* **31**(3) 433–452.
- Zhou, Enlu, Jiaqiao Hu. 2014. Gradient-based adaptive stochastic search for non-differentiable optimization. *IEEE Transactions on Automatic Control* **59**(7) 1818–1832.
- Zhu, Helin, Enlu Zhou. 2016. Risk quantification in stochastic simulation under input uncertainty. $ArXiv \ e-prints: \ http://arxiv.org/abs/1507.06015$.