ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1026

Revisiting the Cross Entropy Method with Applications in Stochastic Global Optimization and Reinforcement Learning

Ajin George Joseph[1] and **Shalabh Bhatnagar**[2]

Abstract. In this paper, we provide a new algorithm for the problem of stochastic global optimization where only noisy versions of the objective function are available. The algorithm is inspired by the well known cross entropy (CE) method. The algorithm takes the shape of a multi-timescale stochastic approximation algorithm, where we reuse the previous samples based on discounted averaging, and hence it saves the overall computational and storage cost. We provide proof of the stability and the global optimization property of our algorithm. The algorithm shows good performance on the noisy versions of global optimization benchmarks and outperforms a state-of-the-art algorithm for non-linear function approximation in reinforcement learning.

1 Introduction and Preliminaries

In this paper, we solve the following problem: For a latent probability measure $\mathbb{P}_{\mathbf{y}}$ over $\mathcal{Y} \subset \mathbb{R}^a$,

$$\text{Find } x^* \in \arg\max_{x \in \mathcal{X} \subset \mathbb{R}^m} L(\mathbb{E}_{\mathbf{y}}\left[J(\mathbf{y})\right], x), \tag{1}$$

where $L : \mathbb{R}^b \times \mathbb{R}^m \rightarrow \mathbb{R}$ is a bounded continuous function and $J : \mathcal{Y} \rightarrow \mathbb{R}^b$ is a bounded function (hence $\mathbb{E}_{\mathbf{y}}[|J|] < \infty$). Here $\mathbb{E}_{\mathbf{y}}[\cdot]$ is the expectation *w.r.t.* $\mathbb{P}_{\mathbf{y}}$. The region $\mathcal{X}$ referred as the *solution space* is a compact subset of $\mathbb{R}^m$. For brevity, we let $\mathcal{H}(x) \triangleq L(\mathbb{E}_{\mathbf{y}}\left[J(\mathbf{y})\right], x)$. In this paper, we assume x^* is unique and $x^* \in interior(\mathcal{X})$.

In this paper, we consider a "black-box" setting, where neither a closed form expression nor structural properties of the objective function $\mathcal{H}$ are available. However, for a given $x \in \mathcal{X}$, a noisy measurement $\bar{\mathcal{H}}(x)$ of the objective function $\mathcal{H}$ is available, where $\bar{\mathcal{H}}(x) = \mathcal{H}(x) + \epsilon(x)$. Here $\epsilon(x)$ is the noise incurred during the measurement of $\mathcal{H}(x)$ which is primarily attributed to the inability to measure accurately the quantity $\mathbb{E}_{\mathbf{y}}[\cdot]$. The intractability of $\mathbb{E}_{\mathbf{y}}[\cdot]$ is due to the hidden probability measure $\mathbb{P}_{\mathbf{y}}$. Problems of the above kind are found in areas of engineering and in discrete-event system simulation [11, 13].

Intuitively, it follows directly from the definition of the problem that any algorithm which solves the problem of the above kind has to be a zero-order method. Since the higher order values of the objective function are unavailable, the algorithm relies only on the function values. These methods are generally called *gradient free* methods. A few prominent algorithms of this kind are simultaneous perturbation stochastic approximation method (SPSA) [26], model reference adaptive search method (MRAS) [8], cross entropy method (CE) [25], estimation of distribution algorithms (EDA) [29] and gradient based adaptive search (GASSO) [30]. The advantage of the above methods is the non-dependency on the structural properties of the objective function and hence can be applied in a generalized setting.

In this paper, we consider the well known cross entropy method. The cross entropy method was motivated by the rare event probability estimation method proposed in [22], which is a variance reduction technique. Later, an iterative procedure based on the above method was applied in various combinatorial optimization problems [23]. Cross entropy method also found successful applications in continuous multi-extremal optimization [21]. A few other applications of the CE method include buffer allocation [1], queuing models [3], DNA sequence alignment [14], control and navigation [6], reinforcement learning [17, 19] and several NP-hard problems [24, 21].

In this paper, we apply a stochastic approximation variant of the CE method to the continuous multi-extremal stochastic optimization problem defined in (1). This particular version of the algorithm is designed to overcome the drawbacks arising due to the inordinate computational and storage requirements of the original CE method. The stochastic approximation nature of our algorithm also helps to adapt naturally to the stochastic optimization setting.

2 The Cross Entropy Method

Notation: We use $\mathbf{x}$ for random variable and x for deterministic variable. Let $\lceil a \rceil$ denote the smallest integer greater than a. Let $supp(f) \triangleq \overline{\{x | f(x) \neq 0\}}$ and $interior(A)$ be the *interior* of set A. Let $f_\theta(\cdot)$ denote the *probability density function* parametrized by θ and $\mathbb{E}_\theta[\cdot]$ be the *expectation w.r.t.* f_θ. Let $\gamma_\rho(\mathcal{H}(\cdot), \theta)$ denote the $(1-\rho)$-*quantile* of $\mathcal{H}(\mathbf{x})$ *w.r.t.* f_θ, *i.e.*,

$$\gamma_\rho(\mathcal{H}(\cdot), \theta) \triangleq \sup\{l : P_\theta(\mathcal{H}(\mathbf{x}) \geq l) \geq \rho\}. \tag{2}$$

$$\text{For } x, y \in \mathbb{R}, \text{ define } \quad \chi(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The *Cross Entropy (CE) method* [25, 9, 4] is a zero-order optimization method to solve optimization problems where the objective function does not posses smooth differentiable structural properties. CE method belongs to a broader class of methods called the *model based search methods*. The goal of the CE method is to find an optimal "*model*" or probability distribution over the solution space $\mathcal{X}$ which concentrates on the global maxima of the objective function. The CE method is an iterative procedure where at each iteration k, a search is conducted on a space of parametrized probability distributions $\{f_\theta | \theta \in \Theta\}$ on $\mathcal{X}$, where Θ is the parameter space, to find a

[1] Indian Institute of Science, Bangalore, email: ajin@csa.iisc.ernet.in
[2] Indian Institute of Science, Bangalore, email: shalabh@csa.iisc.ernet.in

distribution parameter θ_k which reduces the *Kullback-Leibler (KL)*[3] distance from the optimal model. The most commonly used distribution family here is the *exponential family of distributions*.

Exponential Family of Distributions: These distributions are represented by $\mathcal{C} \triangleq \{f_\theta(x) = h(x)e^{\theta^\top \Gamma(x) - K(\theta)} \mid \theta \in \Theta \subset \mathbb{R}^d\}$, where $h : \mathbb{R}^m \longrightarrow \mathbb{R}$, $\Gamma : \mathbb{R}^m \longrightarrow \mathbb{R}^d$ and $K : \mathbb{R}^d \longrightarrow \mathbb{R}$ and $K(\theta) = \log \int h(x)e^{\theta^\top \Gamma(x)}dx$. The Gaussian distribution with mean vector $\mu \in \mathbb{R}^m$ and the covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$ belongs to $\mathcal{C}$. In this case,

$$f_\theta(x) = \frac{1}{\sqrt{(2\pi)^m|\Sigma|}} e^{-(x-\mu)^\top \Sigma^{-1}(x-\mu)/2}, \tag{4}$$

and so one may let $h(x) = 1/(2\pi)^{m/2}$, $\Gamma(x) = (x, xx^\top)^\top$ and $\theta = (\Sigma^{-1}\mu,\ -\frac{1}{2}\Sigma^{-1})^\top$.

Assumption (A1): The parameter space Θ is a compact set.

2.1 CE Method (Ideal Version)

The CE method aims to find a sequence of model parameters $\{\theta_k \in \Theta\}_{k\in\mathbb{Z}^+}$ and an increasing sequence of thresholds $\{\gamma_k \in \mathbb{R}\}_{k\in\mathbb{Z}^+}$ with the property that the support of f_{θ_k} satisfies $supp(f_{\theta_k}) \subseteq \{x|\mathcal{H}(x) \geq \gamma_k\}$. By assigning greater weight to the higher values of $\mathcal{H}$ at each iteration, the expected behaviour of the probability distribution sequence should improve. This is achieved by solving at each instant $k+1$, the following optimization problem: $\theta_{k+1} = \arg\min_{\theta\in\Theta} KL(\hat{f}_{\theta_k}, f_\theta)$, where $\hat{f}_{\theta_k}(x) = \varphi(\mathcal{H}(x))\chi(\mathcal{H}(\mathbf{x}), \gamma_{k+1})f_{\theta_k}(x)$. Further simplification yields,

$$\theta_{k+1} = \arg\max_{\theta\in\Theta} \mathbb{E}_{\theta_k}\left[\varphi(\mathcal{H}(\mathbf{x}))\chi(\mathcal{H}(\mathbf{x}), \gamma_{k+1}) \log f_\theta(\mathbf{x})\right], \tag{5}$$

where $\varphi : \mathbb{R} \to \mathbb{R}_+$ is positive and strictly monotonically increasing. The most common choice for γ_{k+1} is $\gamma_\rho(\mathcal{H}(\cdot), \theta_k)$: the $(1-\rho)$-quantile of $\mathcal{H}(\mathbf{x})$ *w.r.t.* f_{θ_k}, where $\rho \in (0,1)$ is set *a priori*. Also, the parameter space Θ is large enough so that the solution of (5) is contained in $interior(\Theta)$.

We take Gaussian distribution as the preferred choice for $f_\theta(\cdot)$ in this paper. In this case, the model parameter is $\theta = (\mu, \Sigma)^\top$, where $\mu \in \mathbb{R}^m$ is the mean vector and $\Sigma \in \mathbb{R}^{m\times m}$ is the covariance matrix. We obtain a closed-form expression for μ_{k+1} and Σ_{k+1} by equating to 0 the gradient *w.r.t.* $(\Sigma^{-1}\mu,\ -\frac{1}{2}\Sigma^{-1})^\top$ of the objective function in (5) and using (4) for $f_\theta(\cdot)$. We obtain

$$\mu_{k+1} = \frac{\mathbb{E}_{\theta_k}\left[\mathbf{g_1}\big(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{k+1}\big)\right]}{\mathbb{E}_{\theta_k}\left[\mathbf{g_0}(\mathcal{H}(\mathbf{x}), \gamma_{k+1})\right]}, \tag{6}$$

$$\Sigma_{k+1} = \frac{\mathbb{E}_{\theta_k}\left[\mathbf{g_2}\big(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{k+1}, \mu_{k+1}\big)\right]}{\mathbb{E}_{\theta_k}\left[\mathbf{g_0}\big(\mathcal{H}(\mathbf{x}), \gamma_{k+1}\big)\right]}, \tag{7}$$

where $\mathbf{g_0}(\mathcal{H}(x), \gamma) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)$,

$\mathbf{g_1}(\mathcal{H}(x), x, \gamma) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)x$ and

$\mathbf{g_2}(\mathcal{H}(x), x, \gamma, \mu) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)(x-\mu)(x-\mu)^\top$.

Remark 1: Note that in the expression of μ_{k+1} in (6), x is being weighted with $\varphi(\mathcal{H}(x))$ in the region $\{x|\mathcal{H}(x) \geq \gamma_{k+1}\}$. Since the function φ is positive and strictly monotonically increasing, the region where $\mathcal{H}(x)$ is higher (hence $\varphi(\mathcal{H}(x))$ is also higher) is given more weight and hence μ_{k+1} concentrates in the region where $\mathcal{H}(x)$ takes higher values. In case where $\mathcal{H}(\cdot)$ is positive, we can choose $\varphi(x) = x$. However, in general scenarios, where $\mathcal{H}(\cdot)$ takes positive and negative values, the identity function is not an appropriate choice since the effect of the positive weights is reduced by the negative ones. In such cases, we take $\varphi = exp(rx), r \in \mathbb{R}_+$.

[3] The Kullback-Leibler distance between two probability distributions g_1 and g_2 is $KL(g_1, g_2) \triangleq \mathbb{E}_{g_1}\left[\log \frac{g_1}{g_2}\right]$

2.2 CE Method (Monte-Carlo Version)

It is hard in general to evaluate $\mathbb{E}_{\mathbf{y}}[\cdot]$. Hence the objective function values $\mathcal{H}(x)$ might not be available for every value of $x \in \mathcal{X}$. To overcome this, estimates obtained using sample averages are used. The algorithm utilizes a user configured observation allocation rule $\{M_k \in \mathbb{Z}_+\}_{k\in\mathbb{Z}_+}$ to decide the sample size, where $M_k \uparrow \infty$. This means at each iteration k, for a given $x \in \mathcal{X}$, the estimate $\bar{\mathcal{H}}(x)$ is obtained as follows:

$$\bar{\mathcal{H}}(x) = L(\frac{1}{M_k}\sum_{i=1}^{M_k} J(\mathbf{y}_i), x), \text{ where } \mathbf{y}_i \sim \mathbb{P}_{\mathbf{y}}. \tag{8}$$

Also hard to compute are the terms $\mathbb{E}_{\theta_k}[\cdot]$ and γ_{k+1} $(= \gamma_\rho(\mathcal{H}(\cdot), \theta_k))$ in equations (6) and (7). To overcome this, their corresponding stochastic counterparts are employed. Here also we use a user configured observation allocation rule $\{N_k \in \mathbb{Z}_+\}_{k\in\mathbb{Z}_+}$ to decide the sample size, where $N_k \uparrow \infty$. In this Monte-Carlo version, the algorithm generates sequences $\{\bar{\theta}_k = (\bar{\mu}_k, \bar{\Sigma}_k)^\top\}_{k\in\mathbb{Z}_+}$ and $\{\bar{\gamma}_k\}_{k\in\mathbb{Z}_+}$ as follows: At each iteration k, N_k samples $\Lambda_k = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_k}\}$ are chosen using $f_{\bar{\theta}_k}$ and the threshold $\bar{\gamma}_{k+1}$ is obtained as follows:

$$\bar{\gamma}_{k+1} = \bar{\mathcal{H}}_{(\lceil(1-\rho)N_k\rceil)}, \tag{9}$$

where $\bar{\mathcal{H}}_{(i)}$ is the ith-order statistic of $\{\bar{\mathcal{H}}(\mathbf{x}_i)\}_{i=1}^{N_k}$. The model parameter update also uses sample averages. The model parameter $\bar{\theta}_{k+1} = (\bar{\mu}_{k+1}, \bar{\Sigma}_{k+1})^\top$ is updated as follows:

$$\bar{\mu}_{k+1} = \frac{\frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{g_1}(\mathcal{H}(\mathbf{x}_i), \mathbf{x}_i, \bar{\gamma}_{k+1})}{\frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{g_0}(\mathcal{H}(\mathbf{x}_i), \bar{\gamma}_{k+1})}, \tag{10}$$

$$\bar{\Sigma}_{k+1} = \frac{\frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{g_2}(\mathcal{H}(\mathbf{x}_i), \mathbf{x}_i, \bar{\gamma}_{k+1}, \bar{\mu}_{k+1})}{\frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{g_0}(\mathcal{H}(\mathbf{x}_i), \bar{\gamma}_{k+1})}. \tag{11}$$

The Monte-Carlo version is given in Algorithm 1.

Algorithm 1: The Monte-Carlo CE Algorithm

Step 0: Choose an initial *p.d.f.* $f_{\bar{\theta}_0}(\cdot)$ on $\mathcal{X}$, where $\bar{\theta}_0 = (\bar{\mu}_0, \bar{\Sigma}_0)^\top$ and fix an $\epsilon > 0$ (dependent on $\mathcal{H}$).

1. [Sampling Candidate Solutions]: Sample N_k *i.i.d.* solutions $\Lambda_k = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_k}\}$ using $f_{\bar{\theta}_k}$.

2. [Performance Evaluation]: For each $\mathbf{x}$ in Λ_k, take M_k observations $\{\mathbf{y}_i\}_{i=1}^{M_k}$, where $\mathbf{y}_i \sim \mathbb{P}_{\mathbf{y}}$ and calculate the sample average $\bar{\mathcal{H}}(\mathbf{x}) = L(\frac{1}{M_k}\sum_{i=1}^{M_k} J(\mathbf{y}_i), \mathbf{x})$.

3. [Threshold Evaluation]: Calculate the sample $(1-\rho)$-quantile $\bar{\gamma}_{k+1} = \bar{\mathcal{H}}_{(\lceil(1-\rho)N_k\rceil)}$, $\bar{\mathcal{H}}_{(i)}$ is the ith-order statistic of the sequence $\{\bar{\mathcal{H}}(\mathbf{x}_i)\}_{i=1}^{N_k}$.

4. [Threshold Comparison]
if $\bar{\gamma}_{k+1} \geq \bar{\gamma}^*_k + \epsilon$ **then**
  $\bar{\gamma}^*_{k+1} = \bar{\gamma}_{k+1}$,
else
  $\bar{\gamma}^*_{k+1} = \bar{\gamma}^*_k$.

5. [Model parameter update]: Update $\bar{\theta}_{k+1} = (\bar{\mu}_{k+1}, \bar{\Sigma}_{k+1})^\top$ using (10) and (11) with $\bar{\gamma}^*_{k+1}$ instead of $\bar{\gamma}_{k+1}$.

6: If the stopping rule is satisfied, then return $\bar{\theta}_{k+1}$and terminate, else set $k := k+1$ and go Step 1.

2.3 Drawbacks of the CE Method

• *Inefficient use of prior information:* The naive approach of the Monte-Carlo CE does not utilize prior information efficiently. Note that Monte-Carlo CE possesses a stateless behaviour. At each iteration k, a completely new collection of samples are drawn using the distribution f_{θ_k}. The samples are used to derive the estimates $\bar{\gamma}_{k+1}$, $\bar{\mu}_{k+1}$ and $\bar{\Sigma}_{k+1}$. The algorithm does not utilize the estimates generated prior to k.

• *Computational limitations*: These arise due to the dependence on the sample size N_k. One does not know a priori *the best value for the sample size* N_k. Higher values of N_k while resulting in higher accuracy also require more computational resources. One often needs to apply brute force in order to obtain a good choice of N_k. Also as m, the dimension of the solution space $\mathcal{X}$, takes large values, more samples are required for better accuracy, making N_k *large* as well. This makes *finding the ith-order statistic* $\bar{\mathcal{H}}_{(i)}$ *in Step 3 harder*. Note that the order statistic $\bar{\mathcal{H}}_{(i)}$ is obtained by sorting the list $\{\bar{\mathcal{H}}(\mathbf{x}_1), \bar{\mathcal{H}}(\mathbf{x}_2), \ldots \bar{\mathcal{H}}(\mathbf{x}_{N_k})\}$. The computational effort required in that case is $O(N_k \log N_k)$ which is expensive. Also note that N_k diverges to infinity and hence this super linear relationship is computationally very costly.

• *Storage limitations*: The storage requirement for storing sample Λ_k is $N_k * size(\mathbf{x}_k)$. In situations when m and N_k are large, the storage requirement is a major concern.
Note that similar issues are expected to arise with the sample size M_k also.

An illustration in Figure 1 demonstrates the dependency of the performance of Monte-Carlo CE on the sample size schedules $\{N_k\}$ and $\{M_k\}$. We consider here, the Griewank function on $\mathbb{R}^{100}$-*cf.* (36). We take $M_{k+1} = \lceil \alpha_1 M_k \rceil$, $\alpha_1 > 1$ and $N_{k+1} = \lceil \alpha_2 N_k \rceil$, $\alpha_2 > 1$. So a particular schedule can be identified by M_0, α_1 and N_0, α_2. Here we take $\alpha_1 = 1.01$ and $\alpha_2 = 1.005$ for all the schedules, however they differ in their initial values M_0 and N_0. From the Figure 1, note that for $(M_0, N_0) \in \{(100,100), (150,150), (200,200)\}$, the convergence behaviour is close, however for $(M_0, N_0) \in \{(500,500), (1000,1000), (2000,2000)\}$, Monte-Carlo CE converges to a better value. Also when observed carefully, we can notice a significant difference in the limit point of each individual trajectory.

Different variants of the naive Monte-Carlo CE have been proposed in the literature, such as the gradient based Monte-Carlo cross entropy method (GMCCE) [10] and the modified Monte-Carlo cross entropy method [28]. All the variants differ only in the model updating step, the other steps remain the same. Hence they also suffer from the above drawbacks.

3 Proposed Algorithm

In this paper, we resolve the shortcomings of the Monte-Carlo CE method with regards to the concerns mentioned in the previous section, by remodelling the same in the stochastic approximation framework. We follow the same sequence of steps as in Algorithm 1, but differ in their implementation. The stochastic approximation approach streamlines the batch processing of the Monte-Carlo CE. We provide a multi-timescale stochastic approximation algorithm which is efficient, stable, incremental in nature and imposes minimal restriction on the objective function. We avail ourselves of the continuity relationship that holds between the $(1-\rho)$-quantile $\gamma_\rho(\mathcal{H}(\cdot), \theta)$ and the model parameter θ. We also exploit the continuity that holds between μ_{k+1} and Σ_{k+1} *w.r.t.* θ_k. The bootstrapping property inherent in the stochastic approximation algorithms helps to utilize these

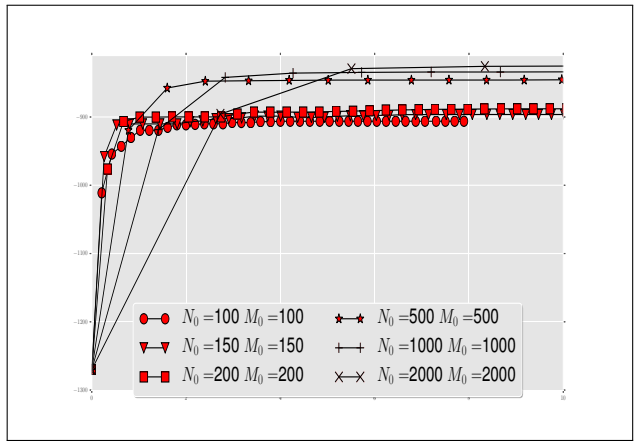


Figure 1: Plot of $\mathcal{H}(\bar{\mu}_k)$, where $\mathcal{H}$ is the Griewank function. The plot shows the dependency of Monte-Carlo CE on the schedules $\{N_k\}$ and $\{M_k\}$.

relationships efficiently.

Stochastic approximation algorithms [2, 15, 20] are a natural way of utilizing prior information. It does so by discounted averaging of the prior information and are usually expressed as recursive equations of the following form:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_{k+1} \Delta \mathbf{z}(\mathbf{z}_k, b_k, D_{k+1})), \tag{12}$$

where $\Delta \mathbf{z}(z, b, D) = h(z) + b + D$ is the increment term, b_k is the bias term with $b_k \to 0$, D_k is a random noise with zero-mean and $h(\cdot)$ a Lipschitz continuous function. The learning rate α_k satisfies $\Sigma \alpha_k = \infty, \Sigma \alpha_k^2 < \infty$.

In our algorithm, we do not apply the naive order statistic to estimate the $(1-\rho)$-quantile. Rather we employ a stochastic recursion which serves the same purpose, but efficiently. To achieve this, we make use of the following lemma from [7]. This lemma provides a characterization of the quantile in terms of an optimization problem.

Lemma 1 *The $(1-\rho)$-quantile of a bounded real-valued function $H(\cdot)$ $\big($with $H(x) \in [H_1, H_2]\big)$ w.r.t the probability distribution $f_\theta(\cdot)$ is reformulated as the optimization problem*

$$\textit{Find } \gamma_\rho(H(\cdot), \theta) = \min_{u \in [H_1, H_2]} \mathbb{E}_\theta \left[\psi(H(\mathbf{x}), u)\right], \tag{13}$$

where $\psi(H(x), u) = (1-\rho)(H(x) - u)\chi(H(x), u) + \rho(u - H(x))\chi(u, H(x))$.

We develop a stochastic gradient recursion which solves the optimization problem in (13). The increment term for the recursion is the sub-differential of ψ *w.r.t.* u, and is given by

$$\nabla_u \psi(H(x), u) = -(1-\rho)\chi(H(x), u) + \rho\chi(u, H(x)). \tag{14}$$

The model parameter update step involves three stochastic recursions. For this we introduce two new variables, $\xi^{(0)}$ and $\xi^{(1)}$, which estimate the RHS of equations (6) and (7) respectively. We also require two increment functions which are defined as follows:

$$\Delta\xi^{(0)}(x, \omega, \gamma) = \mathbf{g_1}(L(\omega, x), x, \gamma) - \xi^{(0)}\mathbf{g_0}(L(\omega, x), \gamma), \tag{15}$$

$$\Delta\xi^{(1)}(x, \omega, \mu, \gamma) = \mathbf{g_2}(L(\omega, x), x, \gamma, \mu) - \xi^{(1)}\mathbf{g_0}(L(\omega, x), \gamma). \tag{16}$$

The algorithm is formally presented in Algorithm 2.

Algorithm 2:

Data: $\alpha_k, \beta_k, \lambda_k, c \in (0,1), \epsilon_1 \in (0,1)$;
Init: $k=0, \gamma_0=0, \gamma_0^*=-\infty, T_0=0, \lambda=\lambda_0, \theta_{old}=NULL$, $\theta_0=(\mu_0,\Sigma_0)^\top, \xi_k^{(0)}=0_{m\times 1}, \xi_k^{(1)}=0_{m\times m}, \omega_0=0_{b\times 1}$;

while *stopping criteria is not satisfied* **do**

$\mathbf{y}_{k+1} \sim \mathbb{P}_{\mathbf{y}}$;

- **[Estimating the objective function]**

$$\omega_{k+1} = \omega_k + \alpha_{k+1}\left(J(\mathbf{y}_{k+1}) - \omega_k\right); \tag{17}$$

- **[Generating the mixture distribution]**

$\widetilde{f}_{\theta_k} = (1-\lambda) f_{\theta_k} + \lambda f_{\theta_0}$;

$\mathbf{x}_{k+1} \sim \widetilde{f}_{\theta_k}$;

- **[Estimating the $(1-\rho)$-quantile]**

$$\gamma_{k+1} = \gamma_k - \beta_{k+1}\nabla_u \psi(L(\omega_k, \mathbf{x}_{k+1}), \gamma_k); \tag{18}$$

- **[Estimating the RHS of equations (6) and (7)]**

$$\xi_{k+1}^{(0)} = \xi_k^{(0)} + \beta_{k+1}\Delta\xi^{(0)}(\mathbf{x}_{k+1}, \omega_k, \gamma_k); \tag{19}$$
$$\xi_{k+1}^{(1)} = \xi_k^{(1)} + \beta_{k+1}\Delta\xi^{(1)}(\mathbf{x}_{k+1}, \omega_k, \xi_k^{(0)}, \gamma_k); \tag{20}$$

if $\theta_{old} \neq NULL$ **then**

$\widetilde{f}_{\theta_{old}} = (1-\lambda) f_{\theta_{old}} + \lambda f_{\theta_0}$; $\quad \mathbf{x}_{k+1}^{old} \sim \widetilde{f}_{\theta_{old}}$;

$$\gamma_{k+1}^* = \gamma_k^* - \beta_{k+1}\nabla_u \psi(L(\omega_k, \mathbf{x}_{k+1}^{old}), \gamma_k^*); \tag{21}$$

- **[Threshold comparison]**

$$T_{k+1} = T_k + c\left(\chi(\gamma_{k+1}, \gamma_{k+1}^*) - \chi(\gamma_{k+1}^*, \gamma_{k+1}) - T_k\right); \tag{22}$$

- **[Updating the model parameters]**

if $T_{k+1} > \epsilon_1$ **then**

$\theta_{old} = \theta_k$;

$$\theta_{k+1} = \theta_k + \alpha_{k+1}\left((\xi_k^{(0)}, \xi_k^{(1)})^\top - \theta_k\right); \tag{23}$$

$$\gamma_{k+1}^* = \gamma_k; \quad T_k = 0; \quad \lambda = \lambda_k; \tag{24}$$

else

$\gamma_{k+1}^* = \gamma_k^*$; $\quad \theta_{k+1} = \theta_k$;

$k := k+1$;

Note that the algorithm uses only 3 samples per iteration: $\mathbf{y}_{k+1}, \mathbf{x}_{k+1}$ and $\mathbf{x}_{k+1}^{old}$. This is a big improvement both computational and storage wise, compared to the original CE method.

Mixture Distribution: In the algorithm, we use a mixture distribution $\widetilde{f}_{\theta_k}$ to generate the sample $\mathbf{x}_{k+1}$, where $\widetilde{f}_{\theta_k} = (1-\lambda)f_{\theta_k} + \lambda f_{\theta_0}$ with λ the mixing weight. λ takes its values from a pre-defined decaying sequence $\{\lambda_k\}_{k\in\mathbb{Z}_+}$, with assignment happening in (24) during the model parameter update step. The initial distribution parameter θ_0 is chosen *s.t.* the density function f_{θ_0} is strictly positive on every point in the solution space $\mathcal{X}$, *i.e.*, $f_{\theta_0}(x) > 0, \forall x \in \mathcal{X}$. The mixture approach facilitates exploration of the solution space and prevents the iterates from getting stranded in suboptimal solutions.

Learning Rates: The learning rates α_k, β_k and the mixing weight λ_k are deterministic, non-increasing and satisfy the following:

$$\lambda_k > 0, \alpha_k > 0, \beta_k > 0, \quad \lim_{k\to\infty}\lambda_k = 0,$$
$$\sum_{k=1}^{\infty}\alpha_k = \sum_{k=1}^{\infty}\beta_k = \infty, \sum_{k=1}^{\infty}\left(\alpha_k^2 + \beta_k^2\right) < \infty, \lim_{k\to\infty}\frac{\alpha_k}{\beta_k} = 0. \tag{25}$$

Since $\alpha_k \to 0$ faster than β_k, the timescale obtained from $\beta_k, k \geq 0$ is faster as compared to the other.

Threshold Comparison Step: The threshold comparison is achieved using the recursion (22) of the random variable T_k. The model parameter θ_k is not updated at each k. Rather it is updated whenever T_k arches over ϵ_1, where $\epsilon_1 \in (0,1)$ is a constant. So the update of θ_k only happens along a subsequence $\{k_{(n)}\}_{n\in\mathbb{Z}_+}$ of $\{k\}_{k\in\mathbb{Z}_+}$. Between $k = k_{(n)}$ and $k = k_{(n+1)}$, the variable γ_k estimates $(1-\rho)$-quantile of $L(\omega_k, \cdot)$ *w.r.t.* $\widetilde{f}_{\theta_{k_{(n)}}}$. The threshold γ_k^* is also updated in (24) during the ϵ_1 crossover. Thus $\gamma_{k_{(n)}}^*$ is the estimate of $(1-\rho)$-quantile *w.r.t.* $\widetilde{f}_{\theta_{k_{(n-1)}}}$.

Notation: We denote by $\gamma_\rho(L(\omega,\cdot),\widetilde{\theta}))$, the $(1-\rho)$-quantile of $L(\omega,\cdot)$ *w.r.t.* the mixture distribution $\widetilde{f}_\theta$ and let $E_{\widetilde{\theta}}[\cdot]$ be the expectation *w.r.t* $\widetilde{f}_\theta$.

Proposition 1: T_k belongs to $(-1,1), \forall k > 0$.

Proof: By rearranging (22) we get,

$$T_{k+1} = (1-c)T_k + c(\chi(\gamma_{k+1}, \gamma_{k+1}^*) - \chi(\gamma_{k+1}^*, \gamma_{k+1})),$$

where $c \in (0,1)$. In the worst case, either $\chi(\gamma_{k+1}, \gamma_{k+1}^*) = 1, \forall k$ or $\chi(\gamma_{k+1}^*, \gamma_{k+1}) = 1, \forall k$. Since the two events are mutually exclusive, we will only consider the former event $\{\chi(\gamma_{k+1}, \gamma_{k+1}^*) = 1, \forall k\}$. In this case

$$\lim_{k\to\infty} T_k = \lim_{k\to\infty}\left(c + c(1-c) + \cdots + c(1-c)^{k-1}\right) = 1.$$

Similarly for $\{\chi(\gamma_{k+1}^*, \gamma_{k+1}) = 1, \forall k\}$, we have $T_k \to -1$. ■

T_k in (22) serves two purposes: first, it is a delay mechanism, whereby it stalls the θ_k-recursion so that $\xi_k^{(0)}$ and $\xi_k^{(1)}$ are sufficiently close to their true values. Second, it ensures that the estimates γ_k eventually become greater than the current threshold $\gamma_{k_{(n)}}^*$, *i.e.*, $\gamma_k > \gamma_{k_{(n)}}^*$ for all but finitely many k.

Remark 2: *The recursion (21) is not addressed in the discussion above. The assignment of γ_{k+1}^* in (24) happens along a subsequence $\{k_{(n)}\}_{n\geq 0}$. Hence $\gamma_{k_{(n)}}^*$ is the estimate of $\gamma_\rho(L(\omega_{k_{(n)}},\cdot),\widetilde{\theta}_{k_{(n-1)}})$. But at time $k_{(n)} < k \leq k_{(n+1)}$, γ_k^* is compared with γ_k in (22). But γ_k is derived from a better estimate of $L(\omega_k,\cdot)$. Equation (21) ensures that γ_k^* is updated using the latest estimate of $L(\omega_k,\cdot)$. The variable θ_{old} holds the model parameter $\theta_{k_{(n-1)}}$ and the update of γ_k^* in (21) is performed using $\mathbf{x}_{k+1}^{old}$ sampled from $\widetilde{f}_{\theta_{old}}$.*

3.1 Convergence Analysis

Assumption (A2): The sequences $\{\omega_k\}_{k\in\mathbb{Z}_+}$ and $\{\gamma_k\}_{k\in\mathbb{Z}_+}$ in (17) and (18) resp. satisfy $\sup_k|\omega_k| < \infty$ and $\sup_k|\gamma_k| < \infty$ *a.s.*.

Remark 3: *The assumption (A2) is a technical requirement to prove convergence of the algorithm. A commonly used procedure to ensure almost sure boundedness of iterates in a stochastic iterative scheme is to project these after each update to an a priori chosen (large enough) compact and convex set. In this case, the bound on the compact set can be derived from the bound on $L(\cdot,\cdot)$.*

Note that the recursion (17) is independent of other recursions and

hence can be analysed independently. For the recursion (17) we have the following result.

Lemma 2 *Let the step-sizes α_k and β_k , $k \in \mathbb{Z}_+$ satisfy (25). Also let (A2) hold. Then the iterates ω_k in (17) satisfy $\omega_k \rightarrow \boldsymbol{\omega}^* = \mathbb{E}_{\mathbf{y}}\left[J(\mathbf{y})\right]$ as $k \rightarrow \infty$ w.p. 1.*

As noted above, the update of θ_k only happens along a subsequence $\{k_{(n)}\}_{n\in\mathbb{Z}_+}$ of $\{k\}_{k\in\mathbb{Z}_+}$. So between $k = k_{(n)}$ and $k = k_{(n+1)}$, θ_k is constant. The lemma and the theorems that follow in this paper depend on the timescale difference in the step-size schedules $\{\alpha_k\}_{k\in\mathbb{Z}_+}$ and $\{\beta_k\}_{k\in\mathbb{Z}_+}$. The step-size $\{\beta_k\}_{k\in\mathbb{Z}_+}$ decays to 0 at a slower rate than $\{\alpha_k\}_{k\in\mathbb{Z}_+}$ and hence the increments in the recursions (18), (19) and (20) which are controlled by β_k are larger and hence converge faster than the recursions (17) and (23) which are controlled by α_k. So the relative evolution of the variables from the slower timescale α_k *i.e.* ω_k, θ_k is slow and can be considered constant when viewed from the faster timescale β_k. See Chapter 6, [2] for a description on multi-timescale algorithms.

Hence, when viewed from the timescale of the recursion (18), one may consider ω_k and θ_k to be fixed. For recursion (18) we have the following result:

Lemma 3 *Let $L(\omega_k, \cdot) \equiv L(\omega, \cdot)$, $\theta_k \equiv \theta, \forall k$. Also let (A2) hold. Then γ_k in (18) satisfy $\lim_{k\rightarrow\infty}\gamma_k = \gamma_\rho(L(\omega,\cdot),\widetilde{\theta})$ w.p. 1 .*

Again, when viewed from the timescale of the recursions (19) and (20), one may consider ω_k and θ_k to be fixed as before. For the recursions (19) and (20), we have the following:

Lemma 4 *Assume $L(\omega_k, \cdot) \equiv L(\omega, \cdot)$, $\theta_k \equiv \theta, \forall k$. Then a.s.,*

$$(i)\ \lim_{k\rightarrow\infty}\xi_k^{(0)} = \xi_*^{(0)} = \frac{\mathbb{E}_{\widetilde{\theta}}\left[\mathbf{g_1}\big(L(\omega,\mathbf{x}),\mathbf{x},\gamma_\rho(L(\omega,\cdot),\widetilde{\theta})\big)\right]}{\mathbb{E}_{\widetilde{\theta}}\left[\mathbf{g_0}\big(L(\omega,\mathbf{x}),\gamma_\rho(L(\omega,\cdot),\widetilde{\theta})\big)\right]}$$

$$(ii)\ \lim_{k\rightarrow\infty}\xi_k^{(1)} = \xi_*^{(1)} = \frac{\mathbb{E}_{\widetilde{\theta}}\left[\mathbf{g_2}\big(L(\omega,\mathbf{x}),\mathbf{x},\gamma_\rho(L(\omega,\cdot),\widetilde{\theta},\xi_*^{(0)})\big)\right]}{\mathbb{E}_{\widetilde{\theta}}\left[\mathbf{g_0}\big(L(\omega,\mathbf{x}),\gamma_\rho(L(\omega,\cdot),\widetilde{\theta})\big)\right]},$$

(iii) If $\gamma_\rho(L(\omega,\cdot),\widetilde{\theta}) > \gamma_\rho(L(\omega,\cdot),\widetilde{\theta}_{old})$, then $T_k, k \geq 0$ in (22) satisfy $\lim_{k\rightarrow\infty} T_k = 1$ a.s.

Notation: For the subsequence $\{k_{(n)}\}_{n>0}$ of $\{k\}_{k\in\mathbb{Z}_+}$, we denote $k_{(n)}^- \triangleq k_{(n)} - 1$ for $n > 0$.

Along the subsequence $\{k_{(n)}\}_{n\geq 0}$ with $k_0 = 0$ the updating of θ_k can be expressed as follows:

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}}\Delta\theta_{k_{(n+1)}}, \tag{26}$$

where $\Delta\theta_{k_{(n+1)}} = (\xi^{(0)}_{k^-_{(n+1)}}, \xi^{(1)}_{k^-_{(n+1)}})^\top - \theta_{k_{(n)}}$. We will prove now that the increment term $\Delta\theta_{k_{(n+1)}}$ in equation (26) is indeed an estimate of $\nabla_{\vartheta(\theta)}\Psi(\theta,\omega^*)\big|_{\theta=\theta_{k_{(n)}}}$, where

$$\boxed{\Psi(\theta,\omega) = \log\mathbb{E}_\theta\left[\mathbf{g_0}\left(L(\omega,\mathbf{x}),\gamma_\rho(L(\omega,\cdot),\theta)\right)\right]} \tag{27}$$

with $\theta = (\mu,\Sigma)^\top$ and $\vartheta(\theta) = (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})^\top$. Before proving this, we state a key lemma about the gradient of Ψ.

Lemma 5 *For the given function $L(\omega,\cdot) \in \mathbb{R}$, $\theta = (\mu,\Sigma)^\top$ and $\vartheta(\theta) = (\vartheta_1,\vartheta_2)^\top = (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})^\top$, we have*

$$\nabla_{\vartheta_1}\Psi(\theta,\omega) = \frac{\mathbb{E}_\theta\left[\mathbf{g_1}\big(L(\omega,\mathbf{x}),\mathbf{x},\gamma_\rho\left(L(\omega,\cdot),\theta\right)\big)\right]}{\mathbb{E}_\theta\left[\mathbf{g_0}\big(L(\omega,\mathbf{x}),\gamma_\rho(L(\omega,\cdot),\theta)\big)\right]} - \mu.$$

$$\nabla_{\vartheta_2}\Psi(\theta,\omega) = \frac{\mathbb{E}_\theta\left[\mathbf{g_2}\big(L(\omega,\mathbf{x}),\mathbf{x},\gamma_\rho\left(L(\omega,\cdot),\theta\right),\mu\big)\right]}{\mathbb{E}_\theta\left[\mathbf{g_0}\big(L(\omega,\mathbf{x}),\gamma_\rho\left(L(\omega,\cdot),\theta\right)\big)\right]} - \Sigma.$$

We now state our main theorem. The following theorem guarantees the convergence of the model sequence $\{\theta_k\}_{k\in\mathbb{Z}_+}$ generated by Algorithm 2 and it further provides a characterization of its limit points. It also shows that by imposing additional structural restrictions on the objective function $\mathcal{H}$, the convergence of the algorithm to the degenerate distribution concentrated on the global maximum x^* is ensured.

Theorem 6 *Let $\varphi(x) = exp(rx), r \in \mathbb{R}$. Assume that the objective function $\mathcal{H}$ satisfies the following two conditions: (i) $\nabla^2\mathcal{H}$ exists and (ii) $\frac{\partial^2\mathcal{H}}{\partial x_i\partial x_j}$ is continuous for $1 \leq \forall i, \forall j \leq m$. Let the learning rates α_k and β_k , $k \in \mathbb{Z}_+$ satisfy (25). Let $\{\theta_k = (\mu_k,\Sigma_k)^\top\}_{k\in\mathbb{Z}_+}$ be the sequence generated by Algorithm 2 and assume $\theta_k \in interior(\Theta)$, $\forall k \in \mathbb{Z}_+$. Also, let (A1) and (A2) hold. Then*

$$\lim_{k\rightarrow\infty}\theta_k = \lim_{k\rightarrow\infty}(\mu_k,\Sigma_k)^\top = (x^*, 0_{m\times m})^\top w.p.1,$$

where x^* is defined in (1).

Proof: Rewriting the equation (23) along the subsequence $\{k_{(n)}\}_{n\in\mathbb{Z}_+}$, we have for $n \in \mathbb{Z}_+$,

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}}\big((\xi^{(0)}_{k^-_{(n+1)}}, \xi^{(1)}_{k^-_{(n+1)}})^\top - \theta_{k_{(n)}}\big). \tag{28}$$

Also $\sup_n\|\theta_{k_{(n)}}\| < \infty$ *a.s.* Rearranging the equation (28) we get, for $n \in \mathbb{Z}_+$,

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}}\left(\mathbb{E}\left[\nabla_{\vartheta(\theta)}\Psi(\theta_{k_{(n)}},\omega^*)\Big|\theta_{k_{(n)}}\right] + o(1)\right), \tag{29}$$

where the $o(1)$ term corresponds to errors in the estimation of $\xi_k^{(0)}$ and $\xi_k^{(1)}$ that decays to zero *a.s.* from Lemma 4.

Now consider the gradient flow ODE

$$\frac{d\theta(t)}{dt} = \nabla_{\vartheta(\theta)}\Psi(\theta(t),\omega^*), \qquad t \in \mathbb{R}_+, \tag{30}$$

where ω^* is defined in Lemma 2.

By appealing to Theorem 2, Chapter 2 of [2], the asymptotic equivalence between the equations (29) and (30) can be easily established. Therefore, the recursion (23) reduces to a stochastic gradient ascent procedure which optimizes the objective function $\Psi(\theta,\omega^*)$. Hence the limiting behaviour of the model sequence $\{\theta_k\}_{k\in\mathbb{Z}_+}$ can be obtained by analysing the same of the above ODE. The fixed points of the ODE (30) can be obtained by equating $\nabla\Psi$ to 0. Equating $\nabla_{\vartheta_1}\Psi(\theta,\omega^*)$ to $0_{m\times 1}$, we get,

$$\mu = \frac{\mathbb{E}_\theta\left[\mathbf{g_1}\big(\mathcal{H}(\mathbf{x}),\mathbf{x},\gamma_\rho(L(\omega^*,\cdot),\theta)\big)\right]}{\mathbb{E}_\theta\left[\mathbf{g_0}\big(\mathcal{H}(\mathbf{x}),\gamma_\rho(L(\omega^*,\cdot),\theta)\big)\right]}. \tag{31}$$

Equating $\nabla_{\vartheta_2}\Psi(\theta,\omega^*)$ to $\mathbb{O}$ $(= 0_{m\times m})$, we get,

$$\frac{\mathbb{E}_\theta\left[\mathbf{g_2}\big(\mathcal{H}(\mathbf{x}),\mathbf{x},\gamma_\rho(L(\omega^*,\cdot),\theta),\mu\big)\right]}{\mathbb{E}_\theta\left[\mathbf{g_0}\big(\mathcal{H}(\mathbf{x}),\gamma_\rho(L(\omega^*,\cdot),\theta)\big)\right]} - \Sigma = \mathbb{O}. \tag{32}$$

To further simplify the notation, let $\gamma_\rho^*(\theta) \triangleq \gamma_\rho(L(\omega^*,\cdot),\theta)$. Also, for brevity let $S(\theta) \triangleq \mathbb{E}_\theta\left[\mathbf{g_0}\big(\mathcal{H}(\mathbf{x}),\gamma_\rho^*(\theta)\big)\right]$ and $\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x},\theta) \triangleq \mathbf{g_0}\big(\mathcal{H}(\mathbf{x}),\gamma_\rho^*(\theta)\big)$. Substituting the expression for μ from (31) in (32) and after further simplification we get,

$$(1/S(\theta))\,\mathbb{E}_\theta\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x},\theta)\mathbf{x}\mathbf{x}^\top\right] - \mu\mu^\top - \Sigma = \mathbb{O}.$$

Since $\Sigma = \mathbb{E}_\theta\left[\mathbf{x}\mathbf{x}^\top\right] - \mu\mu^\top$, the above equation implies

$$\begin{aligned}
&(1/S(\theta))\mathbb{E}_\theta\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x},\theta)\mathbf{x}\mathbf{x}^\top\right] - \mathbb{E}_\theta\left[\mathbf{x}\mathbf{x}^\top\right] = \mathbb{O}\\
\Longrightarrow_1\quad &(1/S(\theta))\mathbb{E}_\theta\left[\left(\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x},\theta) - S(\theta)\right)\mathbf{x}\mathbf{x}^\top\right] = \mathbb{O}\\
\Longrightarrow_2\quad &\Sigma\Sigma\mathbb{E}_\theta\left[\nabla_x^2\mathbf{g_0}(\mathbf{x},\theta)\right] = \mathbb{O}
\end{aligned}$$

$$\implies_3 \quad \Sigma^2 \mathbb{E}_\theta \left[\varphi(\mathcal{H}(\mathbf{x}))G(\mathbf{x})\chi(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))\right] = \mathbb{O}, \quad (33)$$

where $G(x) \triangleq r^2 \nabla\mathcal{H}(x)\nabla\mathcal{H}(x)^\top + r\nabla^2\mathcal{H}(x)$. Note that $\implies_2$ follows from "integration by parts" rule for multivariate Gaussian and $\implies_3$ follows from the assumption $\varphi(x) = exp(rx)$. Note that for each $x \in \mathcal{X}$, $G(x)$ is a $m \times m$ square matrix. Since $(\nabla_i \mathcal{H})^2 \geq 0$, we can find an $r \in \mathbb{R}$ and $1 \leq i \leq m$ *s.t.* $G_{ii}(x) > 0, \forall x \in \mathcal{X}$. This further implies that $\mathbb{E}_\theta[\varphi(\mathcal{H}(\mathbf{x}))G(\mathbf{x})\chi(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))] \neq \mathbb{O}$, $\forall \theta \in \Theta$. Hence, from (33) we get $\Sigma = \mathbb{O}$. This proves that for any $x \in \mathbb{R}^m$, the degenerate distribution concentrated on x given by $\theta_x = (x, 0_{m\times m})^\top$ is an equilibrium point of the ODE (30). Also note that the ODE (30) is asymptotically stable at all local maxima of $\Psi(\cdot, \omega^*)$. The existence of the Lyapunov function $V_x : U_x \to \mathbb{R}_+$ on the open neighbourhood U_x of θ_x, defined as $V_x(\theta) \triangleq \Psi(\theta_x, \omega^*) - \Psi(\theta, \omega^*)$ is enough to prove the local asymptotic stability.

To prove that the limit is indeed θ^*, the degenerate distribution concentrated at x^*, we use *proof by contradiction* technique. So assume to the contrary, *i.e.*, $\theta_k \to \widehat{\theta} = (\widehat{x}, \mathbb{O})^\top$, where $\widehat{x} \neq x^*$. Note that $\mathbb{E}_\theta\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \theta)\right]$, $\mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \theta)\right]$, $\mathbb{E}_\theta\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \theta)\right]$ and $\mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \theta)\right]$ are all continuous on θ. This implies that we can find scalars $\epsilon_2 > 0, \delta_2 > 0$ and $k \in \mathbb{Z}_+$ *s.t.*

$$\begin{aligned}
&\left\|\theta_k - \widehat{\theta}\right\|_\infty < \delta_2, \\
&\left\|\mathbb{E}_{\widehat{\theta}}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \widehat{\theta})\right] - \mathbb{E}_{\theta_k}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \theta_k)\right]\right\|_\infty < \epsilon_2, \\
&\left\|\mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \widehat{\theta})\right] - \mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \theta_k)\right]\right\|_\infty < \epsilon_2, \quad (34) \\
&\left\|\mathbb{E}_{\widehat{\theta}}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \widehat{\theta})\right] - \mathbb{E}_{\theta_k}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \theta_k)\right]\right\|_\infty < \epsilon_2, \\
&\left\|\mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \widehat{\theta})\right] - \mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \theta_k)\right]\right\|_\infty < \epsilon_2,
\end{aligned}$$

where $\|\cdot\|_\infty$ is the sup norm, *i.e.*, $\|x\|_\infty = \max_i |x_i|, x \in \mathbb{R}^m$. Now consider

$$\nabla_{\vartheta_1}\Psi(\theta, \omega^*)|_{\theta=\widetilde{\theta}_k} = (1/S(\theta))\,\mathbb{E}_\theta\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \theta)\right] - \mu\Big|_{\theta=\widetilde{\theta}_k} \quad (35)$$

We denote by $e = (1, \ldots, 1)^\top \in \mathbb{R}^m$. Applying sup norm on either side of (35) and using the inequalities in (34) we get,

$$\begin{aligned}
&\left\|\nabla_{\vartheta_1}\Psi(\theta, \omega^*)|_{\theta=\widetilde{\theta}_k}\right\|_\infty \geq \\
&\left\|\frac{(1-\lambda_k)\mathbb{E}_{\widehat{\theta}}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \widehat{\theta})\right] + \lambda_k \mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{1}(\mathbf{x}, \widehat{\theta})\right] - \epsilon_2 e}{(1-\lambda_k)\mathbb{E}_{\widehat{\theta}}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \widehat{\theta})\right] + \lambda_k \mathbb{E}_{\theta_0}\left[\hat{\mathbf{g}}_\mathbf{0}(\mathbf{x}, \widehat{\theta})\right] + \epsilon_2} - \widehat{x} - \delta_2 e\right\|_\infty \\
&\geq \left\|\frac{(1-\lambda_k)\widehat{x}\varphi(\mathcal{H}(\widehat{x})) + \lambda_k \mathbb{E}_{\theta_0}\left[\mathbf{g}_\mathbf{1}(\mathcal{H}(\mathbf{x}), \mathbf{x}, \mathcal{H}(\widehat{x}))\right] - \epsilon_2 e}{(1-\lambda_k)\varphi(\mathcal{H}(x^*)) + \lambda_k \mathbb{E}_{\theta_0}\left[\mathbf{g}_\mathbf{0}(\mathcal{H}(\mathbf{x}), \mathcal{H}(\widehat{x}))\right] + \epsilon_2}\right. \\
&\qquad \left. - \widehat{x} - \delta_2 e\right\|_\infty \\
&\geq \left\|(K_1(\widehat{x}, \epsilon_2) - 1)\widehat{x} + K_2(\widehat{x}, \epsilon_2)\frac{\mathbb{E}_{\theta_0}\left[\mathbf{g}_\mathbf{1}(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_\rho^*(\widehat{\theta}))\right]}{\mathbb{E}_{\theta_0}\left[\mathbf{g}_\mathbf{0}(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\widehat{\theta}))\right]}\right. \\
&\qquad \left. - (\epsilon_2 + \delta_2)e\right\|_\infty > K_3 > 0,
\end{aligned}$$

where $K_2(\cdot,\cdot) > 0$ and $0 < K_1(\cdot,\cdot) < 1$ with $K_1(x_1, x_2) \to 1$ as $x_1 \to x^*$ and $x_2 \to 0$. This is a contradiction since $\Psi(\theta, \omega^*)$ is continuously differentiable (easily verifiable). ■

4 Experimental Results

For empirical evaluation, we use two settings: (1) Global optimization benchmarks and (2) Nonlinear function approximation setting in reinforcement learning. In each setting, the results shown are averages over 10 independent sample trajectories obtained with the same initial distribution θ_0.

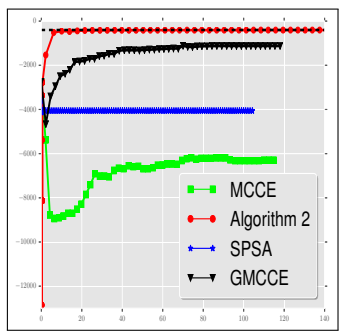
Experiment 1: Global Optimization Benchmarks [12, 5]

We consider here 4 global optimization benchmark functions from [12, 5]. We use their noisy versions to evaluate our algorithm. The function $\varphi(\cdot)$ is chosen as $\varphi(x) = \exp(rx)$, where $r \in \mathbb{R}_+$. We compare the performance of our algorithm with the original Monte-Carlo CE (MCCE), simultaneous perturbation stochastic approximation (SPSA) and the gradient based Monte-Carlo CE (GMCCE) [10] which is a modified version of MCCE. We consider here the noise injected version of SPSA, which is shown to have global optimization properties [18].

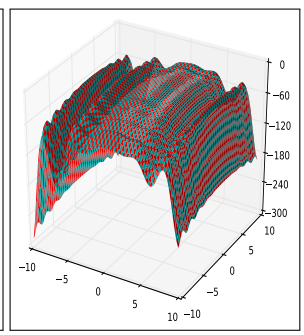
(1) Levy function $[m = 50]$[Continuous, Differentiable]

$$\begin{aligned}
&\mathcal{H}_1(x) = 0.1 * G_1(x)\|\mathbb{E}[Y]\|_2 - \mathbb{E}[Y]^\top \mathbb{E}[Z], \text{ where} \\
&G_1(x) = -1 - \sin^2(\pi y_1) - (y_m - 1)^2(1 + \sin^2(2\pi y_m)) - \\
&\sum_{i=1}^{m-1}[(y_i - 1)^2(1 + 10\sin^2(\pi y_i + 1))] \text{ and } y_i = 1 + \frac{x_i - 1}{4}.
\end{aligned}$$

Here, $Y, Z \in \mathbb{R}^{50}$ with $Y \sim \mathcal{N}([2.0, 2.0, \ldots 2.0]^\top, 40 * I_{50\times 50})$ and $Z \sim \mathcal{N}([4.0, 4.0, \ldots 4.0]^\top, 40 * I_{50\times 50})$. The function has global maximum at the point $x_i = 1$, $1 \leq \forall i \leq m$ with value $\mathcal{H}_1^* = 401.4142135$.



(a) The trajectory of $\mathcal{H}_1(\mu_k)$. The dotted horizontal line is $\mathcal{H}_1^*$. x-axis is the time in secs relative to the start of the algorithm. y-axis is the function value.



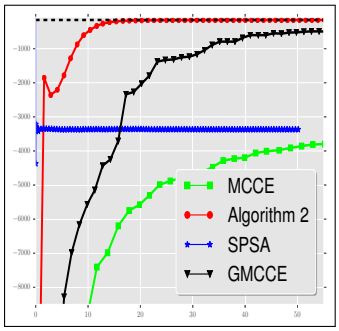
(b) Plot of 3-dimensional $G_1(\cdot)$

Figure 2: Levy Function

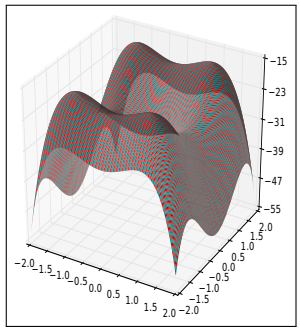
(2) Qing function $[m = 20]$[Continuous, Differentiable, Separable, Scalable, Multimodal]

$$\begin{aligned}
&\mathcal{H}_2(x) = 10^{-3} * G_2(x)\|\mathbb{E}[Y]\|_2 - \mathbb{E}[Y]^\top \mathbb{E}[Z], \\
&\text{where } G_2(x) = -\sum_{i=1}^{m}(x_i^2 - i)^2.
\end{aligned}$$

Here, $Y, Z \in \mathbb{R}^{20}$ with $Y \sim \mathcal{N}([2.0, 2.0, \ldots 2.0]^\top, 40 * I_{20\times 20})$ and $Z \sim \mathcal{N}([4.0, 4.0, \ldots 4.0]^\top, 40 * I_{20\times 20})$.
The function has global maximum at the point $x_i = \sqrt{i}, \forall i, 1 \leq i \leq m$ with value $\mathcal{H}_2^* = -160.0$.

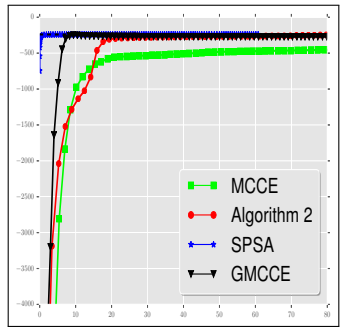


(a) The trajectory of $\mathcal{H}_2(\mu_k)$. The dotted horizontal line is $\mathcal{H}_2^*$. x-axis is the time in secs relative to the start of the algorithm. y-axis is the function value.

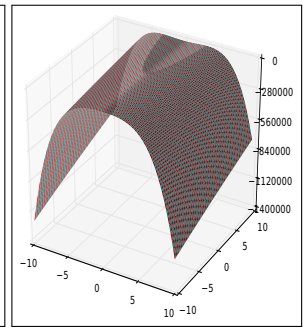


(b) Plot of 3-dimensional $G_2(\cdot)$

Figure 3: Qing Function



(a) The trajectory of $\mathcal{H}_4(\mu_k)$. The dotted horizontal line is $\mathcal{H}_4^*$. x-axis is the time in secs relative to the start of the algorithm. y-axis is the function value.



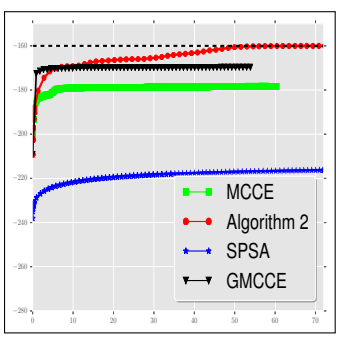
(b) Plot of 3-dimensional $G_4(\cdot)$.

Figure 5: Rosenbrock Function

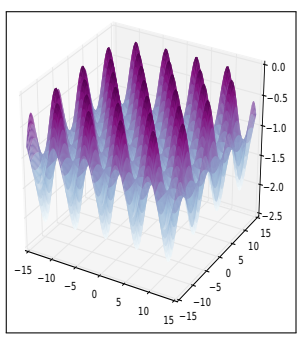
(3) Griewank function $[m = 20]$[Continuous, Differentiable, Non-Separable, Scalable, Multimodal]

$$\mathcal{H}_3(x) = G_3(x)\|\mathbb{E}[Y]\|_2 - \mathbb{E}[Y]^\top \mathbb{E}[Z], \text{ where}$$
$$G_3(x) = -1 - \frac{1}{4000}\sum_{i=1}^{m} x_i^2 + \prod_{i=1}^{m} \cos(x_i/\sqrt{i}). \tag{36}$$

$Y, Z \in \mathbb{R}^{20}$ with $Y \sim \mathcal{N}([2.0, 2.0, \ldots 2.0]^\top, 40 * I_{20\times 20})$ and $Z \sim \mathcal{N}([4.0, 4.0, \ldots 4.0]^\top, 40 * I_{20\times 20})$. The global maximum is $\mathcal{H}_3^* = -160.0$ and is achieved at the origin.



(a) The trajectory of $\mathcal{H}_3(\mu_k)$. The dotted horizontal line is $\mathcal{H}_3^*$. x-axis is the time in secs relative to the start of the algorithm. y-axis is the function value.



(b) Plot of 3-dimensional $G_3(\cdot)$

Figure 4: Griewank Function.

(4) Rosenbrock function$[m = 30]$[Continuous, Differentiable, Non-Separable, Scalable, Unimodal]

$$\mathcal{H}_4(x) = 10^{-6} * G_4(x) - \mathbb{E}[Y]^\top \mathbb{E}[Z],$$

where $G_4(x) = -1 - \sum_{i=1}^{m/2}[100(x_{2i} - x_{2i-1}^2 + (1 - x_{2i-1})^2]$. $Y, Z \in \mathbb{R}^{30}$ with $Y \sim \mathcal{N}([2.0, 2.0, \ldots 2.0]^\top, 10 * I_{30\times 30})$ and $Z \sim \mathcal{N}([4.0, 4.0, \ldots 4.0]^\top, 10 * I_{30\times 30})$. and $\mathcal{H}_4^* = 240.000001$ and is achieved at the point $x_i = 1, 1 \le i \le m$.

To understand the behaviour of our algorithm *w.r.t.* the quantile parameter ρ, we plot the performance of the algorithm for various values of ρ. The results are shown in Figure 6. To demonstrate the advantages of our algorithm with regards to memory utilization, we plot the real time memory usage of our algorithm and GMCCE. The comparison is shown in Figure 7.

Table 1: The parameter values used in experiment 1

$L(\cdot,\cdot)$	α_k	β_k	λ_k	c	ϵ_1	ρ	r
$\mathcal{H}_3$	$\frac{1}{k}$	0.1	$k_{(n)}^{-3.0}$	0.06	0.9	0.1	0.01
$\mathcal{H}_2$	$\frac{1}{k}$	0.09	$k_{(n)}^{-3.0}$	0.06	0.9	0.01	0.002
$\mathcal{H}_4$	$\frac{1}{k}$	0.09	$k_{(n)}^{-3.0}$	0.06	0.9	0.01	0.001
$\mathcal{H}_1$	$\frac{1}{k}$	0.09	$k_{(n)}^{-3.0}$	0.06	0.9	0.01	0.002

From the experiments, we make the following observations:

- Our algorithm shows good performance compared to GMCCE, MCCE and SPSA. Our algorithm also exhibits good global convergence behaviour in all the test cases. These illustrations corroborate the findings of Theorem 6. Note that GMCCE performs better than MCCE in the experiments. The better rate of convergence of GMCCE when compared to MCCE is already shown in [10]. However, the slower rate of these two algorithms when compared to our algorithm is primarily attributed to the insufficient samples. Even though we have chosen reasonably large sample size, it seems insufficient to match the performance of our algorithm. The algorithm also exhibits robustness *w.r.t.* the initial distribution θ_0. An initial distribution which weighs the solution space reasonably well, seems to be sufficient. The values of the parameters λ_k, c and ϵ_1 are same for all test cases. This implies that these parameters require minimal tuning in most cases. However, as with any stochastic approximation algorithm, the choice of the learning rates α_k, β_k is vital.

- We studied the sensitivity of the algorithm with regards to the quantile factor ρ. See Figure 6. For $\rho \in \{0.4, 0.3, 0.2, 0.1\}$, the convergence of the algorithm is fast. However for $\rho \in \{0.5, 0.01, 0.001, 0.0001\}$, the convergence is very slow. Theoretically, the algorithm should converge for all values of ρ. However, in most practical cases, choosing ρ in the range $[0.3, 0.1]$ is highly recommended. Similar observation about the Monte-Carlo CE is mentioned in [8], and this needs further investigation.

- The computational and storage requirements of the algorithm are minimal. See Figure 7. This is attributed to the streamlined and incremental nature of our algorithm. This attribute makes the algorithm suitable in settings where the computational and storage resources are scarce.

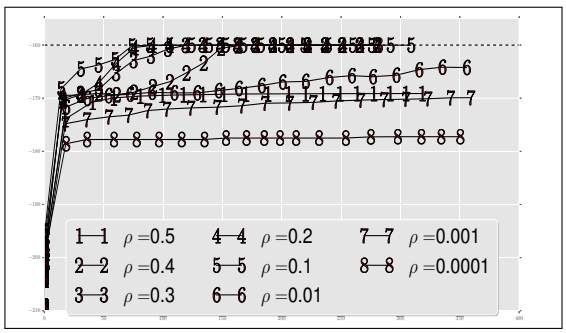


Figure 6: Comparison of our algorithm for various values of ρ.

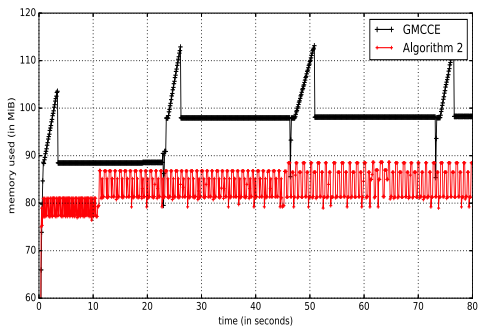


Figure 7: Memory usage comparison.

Experiment 2: Nonlinear Function Approximation for Value Function in Reinforcement Learning [27]

Setup: We consider here a discrete time Markov Chain with state space $\mathbb{S} = \{1, 2, 3\}$, discount factor $\gamma = 0.9$ and transition probability matrix

$$P = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix}.$$

Problem: Given a sample path $\{s_k, r_{k+1}, s_{k+1}, k \geq 0\}$, with s_0 sampled using an arbitrary initial distribution, $s_{k+1} \sim P(s_k, \cdot)$ and the transition payoff $r_{k+1} = 0, \forall k$, solve the optimization problem given by:

$$\lambda^* \in \arg\min_{\lambda \in \mathcal{Q} \subset \mathbb{R}^l} G(\lambda) \triangleq \mathbb{E}\left[\mathbb{E}^2\left[\delta_k\right] | s_k\right], \tag{37}$$

where $\delta_k = r_{k+1} + \gamma V_\lambda(s_{k+1}) - V_\lambda(s_k)$. We also have $V_\lambda(s) = (a(s)\cos(\tau\lambda) - b(s)\sin(\tau\lambda))e^{\epsilon\lambda}$, where $a = [100, -70, -30]^\top$, $b = [23.094, -98.15, 75.056]^\top$, $\tau = 0.01$ and $\epsilon = 0.001$. The true value function[4] is $V = (0, 0, 0)^\top$. The challenge is to best approximate V using the family of nonlinear parameterized functions V_λ by solving the optimization problem (37). It is easy to see that $V_{-\infty} = V$ and hence is a degenerate setting. This particular setting is designed in [27] to show the divergence of the standard TD(0) algorithm in Reinforcement Learning under a nonlinear approximation architecture. A stable non-linear function approximation method called GTD2 [16] converges to the local optima. We believe this is the perfect setting to demonstrate the global convergence property of our algorithm.

[4] Value function $V \in \mathbb{R}^{|\mathbb{S}|}$ and $V(s) = \mathbb{E}[\sum_{k=1}^{\infty} \gamma^k r_{k+1} | s_0 = s]$.

The objective function $G(\cdot)$ in (37) can be rearranged as

$$G(\lambda) = [a_{11}, a_{21}, a_{31}][1.0, 2.0 * e^{\epsilon\lambda}, -2.0 * e^{\epsilon\lambda}]^\top + [e^{\epsilon\lambda}, -e^{\epsilon\lambda}] \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} [e^{\epsilon\lambda}, -e^{\epsilon\lambda}]^\top,$$

where $A = (a_{ij})_{1 \leq i,j \leq 3} \triangleq \mathbb{E}[h_k h_k^\top]$ and

$$h_k = [r_{k+1}, a(s_k) - a(s_{k+1}), b(s_k) - b(s_{k+1})]^\top.$$

The various parameter values we used are as follows:

$\varphi(\cdot)$	α_k	β_k	λ_k	c	ϵ_1	ρ
$\exp(rx), r = 10^{-6}$	$\frac{1}{k}$	0.9	$k_{(n)}^{-3.0}$	0.03	0.95	0.1

The results of the experiment are shown in Figure 8. The x-axis is the iteration number k. The performance measure used is Mean Squared Error (MSE) given by:

$$\text{MSE}(\lambda) = \sum_{i=1}^{i=|\mathbb{S}|} (V(i) - V_\lambda(i))^2 \mu(i), \tag{38}$$

where $\mu(\cdot) \in \mathbb{R}^{|\mathbb{S}|}$ is the stationary distribution of the Markov chain, *i.e.* μ satisfies $\mu^\top P = \mu^\top$.

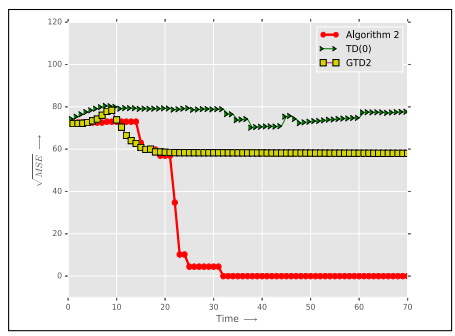


Figure 8: Nonlinear Function Approximation. The plot shows the trajectory of $\sqrt{\text{MSE}}$ generated by TD(0), GTD2 and our algorithm against the iteration number k. **Our algorithm returns the true value function V which follows from the observation that $\sqrt{\text{MSE}}$ converges to 0. TD(0) slowly diverges, while GTD2 converges to a sub-optimal solution**.

5 Conclusion

In this paper, we provided a novel multi-timescale stochastic approximation algorithm for the problem of stochastic optimization, where only noisy versions of the objective function are available. The convergence and the global optimization property of our algorithm are proven using the ODE method. The algorithm shows good performance on noisy versions of benchmark global optimization problems and gives promising results when nonlinear function approximators are used in the setting of Reinforcement Learning.

References

[1] G Alon, Dirk P Kroese, Tal Raviv, and Reuven Y Rubinstein, 'Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment', *Annals of Operations Research*, **134**(1), 137–151, (2005).

[2] Vivek S Borkar, 'Stochastic approximation: A dynamical systems viewpoint', *Cambridge University Press*, (2008).

[3] Pieter-Tjerk de Boer, 'Analysis and efficient simulation of queueing models of telecommunication systems', (2000).

[4] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein, 'A tutorial on the cross-entropy method', *Annals of operations research*, **134**(1), 19–67, (2005).

[5] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger, 'Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions', (2009).

[6] Bjarne E Helvik and Otto Wittner, 'Using the cross-entropy method to guide/govern mobile agents path finding in networks', in *Mobile Agents for Telecommunication Applications*, 255–268, Springer, (2001).

[7] Tito Homem-de Mello, 'A study on the cross-entropy method for rare-event probability estimation', *INFORMS Journal on Computing*, **19**(3), 381–394, (2007).

[8] Jiaqiao Hu, Michael C Fu, and Steven I Marcus, 'A model reference adaptive search method for global optimization', *Operations Research*, **55**(3), 549–568, (2007).

[9] Jiaqiao Hu and Ping Hu, 'On the performance of the cross-entropy method', in *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pp. 459–468. IEEE, (2009).

[10] Jiaqiao Hu, Ping Hu, and Hyeong Soo Chang, 'A stochastic approximation framework for a class of randomized optimization algorithms', *Automatic Control, IEEE Transactions on*, **57**(1), 165–178, (2012).

[11] Deng Huang, Theodore T Allen, William I Notz, and Ning Zeng, 'Global optimization of stochastic black-box systems via sequential kriging meta-models', *Journal of global optimization*, **34**(3), 441–466, (2006).

[12] Momin Jamil and Xin-She Yang, 'A literature survey of benchmark functions for global optimisation problems', *International Journal of Mathematical Modelling and Numerical Optimisation*, **4**(2), 150–194, (2013).

[13] Donald R Jones, Matthias Schonlau, and William J Welch, 'Efficient global optimization of expensive black-box functions', *Journal of Global optimization*, **13**(4), 455–492, (1998).

[14] Jonathan Keith and Dirk P Kroese, 'Rare event simulation and combinatorial optimization using cross entropy: sequence alignment by rare event simulation', in *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, pp. 320–327. Winter Simulation Conference, (2002).

[15] Harold Joseph Kushner and Dean S Clark, *Stochastic approximation methods for constrained and unconstrained systems*, volume 26, Springer Science & Business Media, 2012.

[16] Hamid R Maei, Csaba Szepesvári, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S. Sutton, 'Convergent temporal-difference learning with arbitrary smooth function approximation', in *Advances in Neural Information Processing Systems*, pp. 1204–1212, (2009).

[17] Shie Mannor, Reuven Y Rubinstein, and Yohai Gat, 'The cross entropy method for fast policy search', in *ICML*, pp. 512–519, (2003).

[18] John L Maryak and Daniel C Chin, 'Global random optimization by simultaneous perturbation stochastic approximation', in *American Control Conference, 2001. Proceedings of the 2001*, volume 2, pp. 756–762. IEEE, (2001).

[19] Ishai Menache, Shie Mannor, and Nahum Shimkin, 'Basis function adaptation in temporal difference reinforcement learning', *Annals of Operations Research*, **134**(1), 215–238, (2005).

[20] Herbert Robbins and Sutton Monro, 'A stochastic approximation method', *The Annals of Mathematical Statistics*, 400–407, (1951).

[21] Reuven Rubinstein, 'The cross-entropy method for combinatorial and continuous optimization', *Methodology and computing in applied probability*, **1**(2), 127–190, (1999).

[22] Reuven Y Rubinstein, 'Optimization of computer simulation models with rare events', *European Journal of Operational Research*, **99**(1), 89–112, (1997).

[23] Reuven Y Rubinstein, 'Combinatorial optimization, cross-entropy, ants and rare events', *Stochastic optimization: algorithms and applications*, **54**, 303–363, (2001).

[24] Reuven Y Rubinstein, 'Cross-entropy and rare events for maximal cut and partition problems', *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **12**(1), 27–53, (2002).

[25] Reuven Y Rubinstein and Dirk P Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*, Springer Science & Business Media, 2013.

[26] James C Spall, 'Multivariate stochastic approximation using a simultaneous perturbation gradient approximation', *Automatic Control, IEEE Transactions on*, **37**(3), 332–341, (1992).

[27] John N Tsitsiklis and Benjamin Van Roy, 'An analysis of temporal-difference learning with function approximation', *Automatic Control, IEEE Transactions on*, **42**(5), 674–690, (1997).

[28] Bo Wang and Wayne Enright, 'Parameter estimation for odes using a cross-entropy approach', *SIAM Journal on Scientific Computing*, **35**(6), A2718–A2737, (2013).

[29] Qingfu Zhang and Heinz Mühlenbein, 'On the convergence of a class of estimation of distribution algorithms', *Evolutionary Computation, IEEE Transactions on*, **8**(2), 127–136, (2004).

[30] Enlu Zhou and Jiaqiao Hu, 'Gradient-based adaptive stochastic search for non-differentiable optimization', *Automatic Control, IEEE Transactions on*, **59**(7), 1818–1832, (2014).