Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A Population-Based Cross-Entropy Method with Dynamic Sample Allocation

Jiaqiao Hu and Hyeong Soo Chang

Abstract— **This paper generalizes the cross-entropy (CE) method to a population-based setting, where a *population* of probabilistic models is maintained/updated and subsequently propagated from generation to generation. One of the key questions in the proposed approach is how to efficiently distribute a given sample budget among different models in a population to maximize algorithm performance. We formulate this problem as a Markov decision process (MDP) model and derive an efficient dynamic sample allocation scheme to adaptively allocate computational resources. We carry out numerical studies to illustrate the method and compare its performance with existing procedures.**

I. Introduction

We consider deterministic global optimization problems over real vector-valued domains. The general solution methods to solve these problems can be broadly classified into two categories (cf. e.g., [13]): the *instance-based* methods and the *model-based* search (MBS) methods. In instance-based methods, the search for new candidate solutions depends directly on previous generated solutions. Some well-known instance-based methods are simulated annealing (SA) [6], genetic algorithms (GAs) [11], tabu search [4], the adaptive partitioned random search (APRS) [12], and the nested partitions (NP) method [10]. On the other hand, in model-based methods, new solutions are generated via an intermediate probability model over the solution space that is iteratively updated. Thus, each iteration of a model-based algorithm usually involves a solution generation step, which randomly samples candidate solutions from the current probability model, and a model updating step, which updates the current probability model based on the generated solutions to bias the future search toward regions containing high quality solutions. Some examples of model-based methods are ant colony optimization (ACO) [2], estimation of distribution algorithms (EDAs) [7], the cross-entropy (CE) method [9], and the recently proposed model reference adaptive search (MRAS) [5], see e.g., [3] and [13] for a recent review.

In this paper, we focus on the CE method, and propose a heuristic extension we call Population-based CE with Dynamic Sample Allocation (PCEDSA) that aims to improve both the efficiency and robustness of the original CE method by maintaining and updating a *population* of probabilistic models rather than just a *single* model. This generalization has given rise to a key question as to how to allocate a given sample budget (function evaluations) to a population of distribution models so that the expected performance of the algorithm is optimized in some sense. We study this question from a decision-theoretic point of view, and formulate the problem as a Markov decision process (MDP) model with a terminal reward function. However, because obtaining the exact optimal policy for the MDP model is intractable, a sub-optimal one-step look-ahead policy that myopically optimizes at each step a lower bound of the optimal value function is derived. The sample allocation policy dynamically assigns to each model in the population a performance index and suggests that the next distribution model to sample from is the one with the current best performance index. These performance indices are then further used in conjunction with the CE method to update the current population to obtain a new population of distribution models. We show that the new generation of distributions improves upon the previous generation in the sense that each of the models in the new generation has a better performance index than its corresponding predecessor.

The rest of the paper is organized as follows. In Section II, we begin with the problem setting and derive a sample allocation policy to adaptively distribute computational resources. We give a detailed description of the algorithm in Section III, where the idea of Section II is efficiently combined with the CE method to update the distribution population. Some implementation issues of PCEDSA are discussed in Section IV and numerical examples are given in Section V. Finally we conclude this paper in Section VI.

II. Dynamic Sample Allocation

A. Background

Consider solving the global optimization problem

$$x^* \in \arg\max_{x \in \mathbb{X}} H(x), \tag{1}$$

where the solution space $\mathbb{X}$ is a non-empty set in $\Re^n$, and $H(\cdot) : \mathbb{X} \to \Re$ is a deterministic real-valued function. We assume that problem (1) has at least one global maximizer x^* that achieves the maximum of H. Note that the function H is not necessarily convex or continuous, and there could be many local optimal solutions.

In this paper, we approach this problem by using an evolutionary, population- and model-based random sampling approach, i.e., by explicitly specifying a set or population of potentially good probability distribution models, and then iterating on this set to produce improving models. In particular, let $\Lambda_f = \{f_{\theta_1}, \ldots, f_{\theta_K}\}$ be a set of K parameterized

J. Hu is with the Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794, USA jqhu@ams.sunysb.edu

H. S. Chang is with the Department of Computer Science and Engineering, Sogang University, Seoul, Korea hschang@sogang.ac.kr

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

probability density/mass functions (p.d.f./p.m.f.) on $\mathbb{X}$. We assume that $f_{\theta_i},\ i = 1,\ldots,K$ belong to some common parameterized distribution family $\{f_\theta,\ \theta \in \Theta\}$, where Θ is the parameter space. We remark that the use of parameterized distributions is mainly of practical convenience, as they can generally be efficiently sampled from and represented relatively compactly by their associated parameters. The general outline of the approach consists of the following two steps: at each iteration k,

1) Sample N candidate solutions from the population $\Lambda_f^k := \{f_{\theta_1^k},\ldots,f_{\theta_K^k}\}$ and evaluate the objective function value at each of those points;
2) Iterate on the current population Λ_f^k by using the N solutions collected in the first step to produce an improving population of distributions $\Lambda_f^{k+1} := \{f_{\theta_1^{k+1}},\ldots,f_{\theta_K^{k+1}}\}$.

The above procedure resembles that of a typical model-based algorithm, but generalizes it by allowing candidate solutions to be generated from K different distributions. This generalization immediately leads to the following two questions: (1) how to allocate the N random samples to the K distributions at each iteration of the algorithm? (2) how to update the population of distributions in a synergetic way to obtain an improving population?

Throughout this paper, we use $P_{\theta_i}(\cdot)$ and $E_{\theta_i}[\cdot]$ to denote the probability and expectation taken with respect to f_{θ_i}, and $I\{\cdot\}$ to denote the indicator function.

B. Dynamic Sample Allocation Rule

In this section, we formulate the sample allocation problem as a Markov decision process (MDP) model and derive an efficient decision rule for that problem.

Given a total of N samples (function evaluations), at step $t = 0$, we start by generating a small number of i.i.d. samples, say m, from each of K distribution models, where $mK \le N$. We denote by $\Lambda_X^0 = \{X_0^1, X_0^2, \ldots, X_0^{mK}\}$ the set of mK samples generated and $\Lambda_H^0 = \{H_0^1, H_0^2, \ldots, H_0^{mK}\}$ the set of corresponding function values evaluated at those points, i.e., $H_0^i = H(X_0^i)$ for all $i = 1,\ldots,mK$. We assume that both Λ_X^0 and Λ_H^0 are *multisets* so that they may contain the same element more than once. Order the mK function values from smallest to largest, i.e., $H_0^{(1)} \le H_0^{(2)} \le \cdots \le H_0^{(mK)}$, and let $\overrightarrow{\Lambda_H^0} \in \Re^{1\times mK}$ be a row vector whose ith entry is given by the ith order statistic $H_0^{(i)}$. For a given integer $0 < L \le mK$, we also define $\gamma_0^\ell = \overrightarrow{\Lambda_H^0}(mK - L + \ell)$ for all $\ell = 1,\ldots,L$. In other words, $\gamma_0^\ell,\ \ell = 1,\ldots,L$ are the L largest function values obtained out of mK samples. For example, γ_0^1 is the Lth largest function value and γ_0^L is the largest function value.

At each step $t = 1,\ldots,N-mK$, exactly one candidate solution will be generated from one of the K distributions. Let H_t be the function value corresponding to that candidate solution, and let $\{\gamma_t^\ell,\ \ell = 1,\ldots,L\}$ be the set of largest L function values obtained at step t. We define a vector $W_t = (\gamma_t^1,\ldots,\gamma_t^L, n_t)$, where n_t is a counter that counts the total number of improvements in the set of best L function values obtained up to time t, i.e., $n_t = n_{t-1} + I\{H_t \ge \gamma_{t-1}^1\}$ with $n_0 = 0$.

Now consider a (possibly non-stationary) sample allocation rule/policy π which determines, based only on W_t, at each step t, whether at least one more sample should be taken or the entire sampling process should be terminated; in the former case, the policy π also specifies which of K distribution models should be sampled from and then takes exactly one sample from that distribution. We let $\pi_t(W_t)$ be the decision or action chosen by policy π at step t, which takes values from the set $A = \{0,1,\ldots,K\}$, where

$$\pi_t(W_t) := \begin{cases} 0 & \text{stop sampling at step } t, \\ i & \text{take one sample from } f_{\theta_i},\ i = 1,\ldots,K. \end{cases}$$

When $\pi_t(W_t) \ne 0$, we let $X_{t+1}^{\pi_t(W_t)}$ be the sample obtained at step $t+1$, and define $\Lambda_X^{t+1} = \Lambda_X^t \cup \{X_{t+1}^{\pi_t(W_t)}\}$, $\Lambda_H^{t+1} = \Lambda_H^t \cup \{H(X_{t+1}^{\pi_t(W_t)})\}$, and $\gamma_{t+1}^\ell = \overrightarrow{\Lambda_H^{t+1}}(mK+t+1-L+\ell)$ for all $\ell = 1,\ldots,L$ and $t = 0,\ldots,N-mK-1$, where $\overrightarrow{\Lambda_H^{t+1}}$ is constructed in the same way as described above. We also define a termination state $\mathcal{T}$, and whenever the stop action is chosen at step t, i.e., $\pi_t(W_t) = 0$, we define $n_{r+1} = \mathcal{T}$ and $\gamma_{r+1}^\ell = \mathcal{T}$ for all $\ell = 1,\ldots,L$ and $r = t,\ldots,N-mK-1$.

Lemma 2.1: For every sample allocation policy π described above, the process $\{W_t\}$ is a Markov chain with absorbing state $\mathscr{T}$, where $\mathscr{T}$ is an 1-by-$(L+1)$ vector with all elements equal to $\mathcal{T}$, i.e., $\mathscr{T} = (\mathcal{T},\mathcal{T},\ldots,\mathcal{T})$.

Proof: It is easy to verify that the following recursion holds whenever $\pi_t(W_t) \ne 0$: for all $\ell = 1,\ldots,L-1$,

$$\begin{aligned} \gamma_{t+1}^\ell &= \gamma_t^\ell + \big(H(X_{t+1}^{\pi_t(W_t)}) - \gamma_t^\ell\big)^+ \\ &\qquad - \big(H(X_{t+1}^{\pi_t(W_t)}) - \gamma_t^{\ell+1}\big)^+, \\ \gamma_{t+1}^L &= \gamma_t^L + \big(H(X_{t+1}^{\pi_t(W_t)}) - \gamma_t^L\big)^+ \ \text{ and} \\ n_{t+1} &= n_t + I\{H(X_{t+1}^{\pi_t(W_t)}) \ge \gamma_t^1\}, \end{aligned} \tag{2}$$

for all $t = 0,\ldots,N-mK-1$, where $Z^+ := \max\{0,Z\}$. Thus, the claim follows by the construction of $\{W_t\}$. ∎

Now consider the process $\{W_t\}$ with some associated cost, in which when in state $W_t = w$ at step t we may either stop sampling (i.e., $\pi_t(W_t) = 0$) and receive a termination reward n_t, or choose to generate a sample from the distribution f_{θ_i} at the expense of paying a cost $C \in [0,1)$. Note that C may also include the function evaluation cost at the sampled point. Thus, the above becomes an $(N-mK+1)$-horizon Markov decision process with the following one-stage-reward function

$$R_t(w,a) := \begin{cases} n_t & \text{if } a = 0, \\ -C & \text{if } a = 1,\ldots,K. \end{cases} \tag{3}$$

Thus, for a given sample allocation policy π and an initial state $W_0 = w$, the total reward accumulated before termination is given by

$$V^\pi(w) = E\Big[\sum_{t=0}^{N-mK} R_t\Big(W_t, \pi_t(W_t)\Big)\Big| W_0 = w\Big], \tag{4}$$

where $E[\cdot]$ is understood with respect to the probability law

induced by policy π. The objective is to find an optimal sample allocation policy π^* to maximize the total reward accumulated before termination for all given initial states. In view of (3) and (4), it is easy to see that we are essentially maximizing, less the sampling and function evaluation cost, the expectation of the total number of improvements that can be achieved before the process terminates.

We define the set

$$\mathscr{S} = \Big\{ w : \max_{1\le i\le K} \big\{ P_{\theta_i}\big(H(X) \ge w(1)\big) - C \big\} \le 0 \Big\}, \quad (5)$$

where $w(1)$ indicates the first component of the state w, i.e., the current Lth best function value obtained. Roughly speaking, $\mathscr{S}$ is the set of states so that the maximum probability of generating a solution that is at least as good as the current Lth best solution is less than the cost of generating such a solution. Let V_t be the optimal reward function of the $(N - mK - t)$-stage tail problem of the original MDP. The following theorem provides a lower bound to the optimal value function.

Theorem 1: For any state $w \neq \mathscr{T}$, the optimal sample allocation policy π^* satisfies $\pi_t^*(w) = 0$ whenever $w \in \mathscr{S}$. Moreover

$$V_t(w) \begin{cases} = w(L+1) & \text{if } w \in \mathscr{S} \\ \ge w(L+1) & \text{if } w \notin \mathscr{S}, \end{cases}$$

for all $t = 0, 1, \ldots, N - mk$, where recall that $w(L+1)$ is the $L+1$th element of w.

Note that when $C = 0$, $\mathscr{S}$ becomes an empty set, in which case, Theorem 1 suggests that it is never optimal to stop sampling until all the available sample budget is exhausted.

Proof: Follows from the dynamic programming algorithm (e.g., [1]). ■

Unfortunately, since obtaining the exact optimal sample allocation policy is intractable even when the horizon length is small, we focus on the one-step look-ahead sample allocation policy given by

$$\hat{\pi}_t(w) = \begin{cases} 0 & \text{if } w \in \mathscr{S} \\ \arg\max\limits_{1\le i\le K} \big\{ P_{\theta_i}(H(X) \ge w(1)) \big\} - C & \text{if } w \notin \mathscr{S}. \end{cases}$$

At each time t, it is easy to observe that $\hat{\pi}_t(w)$ allocates the next sample to myopically maximize the expected improvement $E_{\theta_i}[n_{t+1}]$ after the additional allocation. From the proof of Theorem 1, this is essentially equivalent to maximizing at each stage the sum of the current one-stage reward and the lower bound of the optimal reward-to-go given in the theorem. Note that the allocation policy $\hat{\pi}_t(w)$ is in fact stationary (i.e., independent of t). This feature makes the allocation policy convenient to implement. Consequently, for any given observation $w(1) = \gamma$, if we associate to each distribution model f_{θ_i} a performance index

$$\mathcal{I}_i(\gamma) := P_{\theta_i}(H(X) \ge \gamma),$$

then the policy suggests that if at least one more sample is needed, then that sample should be generated from the distribution having the current best performance index.

III. Population-Based CE with Dynamic Sample Allocation

We now describe the PCEDSA algorithm that uses the dynamic sample allocation idea to adaptively distribute computational resources in a population-base setting. A high-level description of the algorithm is provided in Figure 1, where for ease of exposition, we have assumed that $C = 0$, i.e., no early stopping can occur.

PCE with Dynamic Sample Allocation

- **Initialization:** Specify $N > 0$ allowed at each iteration and an initial population $\Lambda_f^0 = \{f_{\theta_1^0}, \ldots, f_{\theta_K^0}\}$. Choose the algorithm parameters $L \ge 1$ and $m \ge 1$ with $mK \le N$. Set iteration counter $k = 0$.
- **Repeat until a specified stopping rule is satisfied:**
  - **Dynamic Sample Allocation:**
    - Generate m i.i.d. samples from $f_{\theta_i^k}$ for all $i = 1, \ldots, K$. Denote the set of generated solutions and their function values by $\Lambda_X = \{X_0^1, \ldots, X_0^{mK}\}$ and $\Lambda_H = \{H(X_0^1), \ldots, H(X_0^{mK})\}$, respectively. Set $\gamma_0 = \overrightarrow{\Lambda_H}(mK - L + 1)$. Calculate performance indices $\mathcal{I}_i(\gamma_0)$ for all $i = 1, \ldots, K$, and let
      $$i^* = \arg\max_{1\le i\le K} \mathcal{I}_i(\gamma_0).$$
    - for $t = 0$ **to** $N - mK - 1$
      Take one sample X_{t+1} from $f_{\theta_{i^*}^k}$. Set
      $\Lambda_X = \Lambda_X \cup \{X_{t+1}\}$,
      $\Lambda_H = \Lambda_H \cup \{H(X_{t+1})\}$, and
      $\gamma_{t+1} = \overrightarrow{\Lambda_H}(mK + t + 2 - L)$.
      Calculate performance indices $\mathcal{I}_i(\gamma_{t+1})$ for all $i = 1, \ldots, K$.
      Let $i^* = \arg\max_{1\le i\le K} \mathcal{I}_i(\gamma_{t+1})$.
      end
  - **Distribution Population Construction:**
    - Update all K distributions individually to obtain the next generation $\Lambda_f^{k+1} = \{f_{\theta_1^{k+1}}, \ldots, f_{\theta_K^{k+1}}\}$.
    - $k \leftarrow k + 1$.

Fig. 1. A high-level description of PCEDSA

We start by specifying a computational budget N, the initial number of samples m to be generated from each distribution model, and a constant $L \ge 1$. As we will see in Section IV, L is the number of candidate solutions that will be used in updating the distribution models. Once chosen, these parameters are fixed throughout the algorithm. We then select an initial population of distributions. In practice, the initial population can be chosen according to some prior knowledge of the problem structure; however, in cases where no such information is available, one simple choice is to uniformly/equally distribute the K distributions over the solution space so that each subregion will have a good chance of being sampled. For example, one possibility is to roughly partition the solution space into K equal subregions, and then place a distribution over each of these K regions.

A. Distribution Population Construction

The overall idea of distribution construction in PCEDSA is based on the intuition that once a set of good sampling

distributions have been identified based on the past sampling history, we should exploit this fact and concentrate more future computational effort on these distributions. However, in practice, persistent improvements will be increasingly difficult to achieve as the number of sampling instances increases, since the performance index $\mathcal{I}(\gamma)$ is a decreasing function of γ, which implies that the probability of finding improving solutions becomes smaller. This suggests that we should find a viable way to constantly update these distribution models so that their corresponding performance indices can be improved from time to time.

In PCEDSA, an individual distribution model is updated by using the CE method. In an optimization context, the basic idea of CE is to work with a parameterized distribution on the solution space and try to find an optimal parameter of the distribution that assigns maximum probability to the set of near optimal solutions. For a detailed description, the reader is referred to [9]. An alternative interpretation of CE is also given in [5], where it has been shown that there is a sequence of intermediate distribution models referred to as *reference* models implicit in CE, and each step of the algorithm minimizes the Kullback-Leibler (KL) distance between the parameterized distribution and the *reference* distribution. The above rationale has resulted in the following distribution updating procedure.

Distribution Population Updating:

Upon termination of the **Dynamic Sample Allocation** step of the algorithm, we obtain a random variable γ_{N-mK}, which is the Lth best function value among the total N samples obtained. Given the current observation $\gamma_{N-mK} = \gamma$, define *reference* distributions

$$g_i^{k+1}(x) = (1-\alpha)\frac{I\{H(x) \geq \gamma\} f_{\theta_i^k}(x)}{\mathcal{I}_i(\gamma)} + \alpha f_{\theta_i^k}(x) \quad (6)$$

for all $i = 1, \ldots, K$, where α is a specified constant between 0 and 1.

- Update the parameters associated with the K distributions by minimizing the KL-distance

$$\theta_i^{k+1} = \arg\min_{\theta \in \Theta} \mathscr{D}\big(g_i^{k+1},\ f_\theta\big) \quad \text{for all } i = 1, \ldots, K. \quad (7)$$

- Set the next population $\Lambda_f^{k+1} = \{f_{\theta_1^{k+1}}, \ldots, f_{\theta_K^{k+1}}\}$.

In Equation (7), the KL distance between two probability distribution g and f is defined as follows:

$$\mathscr{D}(g, f) = \int_{x \in \mathbb{X}} \ln \frac{g(x)}{f(x)} f(x) \nu(dx), \quad (8)$$

where $\nu(\cdot)$ is the Lebesgue/discrete measure on $\mathbb{X}$. We note that the sequence of *reference* distributions g_i^{k+1} is only used *implicitly* to guide the parameter updating procedure and there is no need to build them explicitly at each iteration of the algorithm. Clearly, the above distribution updating procedure follows that of the standard CE method. Consequently, it is easy to see that PCEDSA degenerates to the standard CE method when the population size $K = 1$.

We now present the main result of this section, which states that the performance indices of the K updated distributions are improving. We begin with some regularity conditions.

Assumptions:

A1. *The parameterized density/mass functions* $f_\theta,\ \forall\, \theta \in \Theta$ *have the same support.*

A2. $f_{\theta_1}(x) \neq f_{\theta_2}(x)$ *almost surely for all* $\theta_1 \neq \theta_2$.

Theorem 2: For a given $\gamma < H(x^*)$, let $\mathcal{I}_i^*(\gamma)$ be the performance index of the ith distribution obtained via Equation (7) for all $i = 1, \ldots, K$. If Assumption A1 is satisfied, then the K new distributions generated by the "Distribution Population Construction" step of PCEDSA are improving, in the sense that

$$\mathcal{I}_i^*(\gamma) \geq \mathcal{I}_i(\gamma) \quad \text{for all } i = 1, \ldots, K.$$

Moreover, if Assumption A2 is also satisfied and $\theta_i^{k+1} \neq \theta_i^k$, then

$$\mathcal{I}_i^*(\gamma) > \mathcal{I}_i(\gamma) \quad \text{for all } i = 1, \ldots, K.$$

Proof: From Equation (7), since θ_i^{k+1} minimizes the KL distance $\mathscr{D}(g_i^{k+1}, f_\theta)$, we have

$$\begin{aligned} 0 &\leq \mathscr{D}\big(g_i^{k+1}, f_{\theta_i^k}\big) - \mathscr{D}\big(g_i^{k+1}, f_{\theta_i^{k+1}}\big) \\ &= E_{g_i^{k+1}}\Big[\ln \frac{f_{\theta_i^{k+1}}(X)}{f_{\theta_i^k}(X)}\Big] \leq E_{g_i^{k+1}}\Big[\frac{f_{\theta_i^{k+1}}(X)}{f_{\theta_i^k}(X)} - 1\Big], \end{aligned}$$

where $E_{g_i^{k+1}}[\cdot]$ is the expectation taken with respect to g_i^{k+1}, and the last inequality follows because $\ln x \leq x - 1$ for all $x > 0$. Consequently, it follows from the definition of g_i^{k+1} (cf. Equation (6)) that

$$1 \leq E_{g_i^{k+1}}\Big[\frac{f_{\theta_i^{k+1}}(X)}{f_{\theta_i^k}(X)}\Big] = (1-\alpha) E_{\theta_i^{k+1}}\Big[\frac{I\{H(X) \geq \gamma\}}{\mathcal{I}_i(\gamma)}\Big] + \alpha$$

Therefore, we have

$$\mathcal{I}_i(\gamma) \leq E_{\theta_i^{k+1}}\big[I\{H(X) \geq \gamma\}\big] = \mathcal{I}_i^*(\gamma). \quad (9)$$

Note that (9) becomes equality if and only if $f_{\theta_i^{k+1}}(x) = f_{\theta_i^k}(x)$ almost surely on $\mathbb{X}$. Thus, if Assumption A2 is satisfied, then (9) becomes strict inequality. Hence, the second claim in Theorem 2 follows. ■

Theorem 2 also implies that the expected performance of the original CE method with smoothed parameter updating is improving, in the sense that the probability of generating an improving solution (i.e., a solution whose function value is better than γ) under the new distribution $f_{\theta_i^{k+1}}$ is greater than the probability of generating an improving solution under the old distribtuion $f_{\theta_i^k}$.

IV. Implementation Issues

In this section, we consider the problems of estimating performance indices and updating distribution models based on the actual observed data.

A. Estimating Performance Index

The key question to address is how to use a single sample trajectory (i.e., the N samples generated by the sample

allocation policy) to efficiently estimate the performance indices of all K distribution models.

Here, we propose an approach that is based on the importance sampling idea. Specifically, let Λ_X^i be the set of candidate solutions sampled from distribution f_{θ_i}, and define $\bar{\Lambda}_X^i := \Lambda_X^i \cap \{x \in \Lambda_X : H(x) \geq \gamma_{N-mK}\}$, i.e., $\bar{\Lambda}_X^i$ is the set of samples generated from f_{θ_i} with performance better than γ_{N-mK}.

Lemma 4.1: For any $\gamma_{N-mk} := \gamma < H(x^*)$, if Assumption A1 is satisfied, then

$$\hat{\mathcal{I}}_i(\gamma) = \frac{1}{N}\sum_{j=1}^{K}\sum_{x\in\bar{\Lambda}_X^j}\frac{f_{\theta_i}(x)}{f_{\theta_j}(x)} \tag{10}$$

is an unbiased estimator of $\mathcal{I}_i(\gamma)$ for all $i = 1, \ldots, K$.

Proof: The result follows from Wald's identity [8] because $|\Lambda_X^i|$ (the cardinality of the set Λ_X^i) depends only on the past sampling history. ∎

B. Updating Distribution Models

We focus our discussion on a particular family of distributions called the Natural Exponential Family (NEF). For such a distribution family, it has been shown (cf. e.g., [9]) that the optimization problem (7) can be solved analytically in closed form.

Definition 1: A parameterized family of p.d.f's/p.m.f's $\{f_\theta,\ \theta \in \Theta \subseteq \Re^m\}$ on $\mathbb{X}$ is said to belong to the natural exponential family (NEF) if there exist functions $h(\cdot) : \Re^n \to \Re$, $c(\cdot) : \Re^n \to \Re^m$, and $\Upsilon(\cdot) : \Re^m \to \Re$ such that

$$f_\theta(x) = \exp\left\{\theta^T c(x) - \Upsilon(\theta)\right\} h(x), \quad \forall\, \theta \in \Theta, \tag{11}$$

where $\Upsilon(\theta) = \ln \int_{x\in\mathbb{X}} \exp\left\{\theta^T c(x)\right\} h(x)\nu(dx)$, and the superscript "$T$" denotes the vector transposition.

The following result generalizes Lemma 2 in [5].

Lemma 4.2: For any given distribution function g, let $\theta^* = \arg\min_{\theta\in\Theta} \mathscr{D}(g, f_\theta)$. Assume that

1) $\sup_{\theta\in\Theta} \|e^{\theta^T c(x)} c(x) h(x)\|$ is integrable/summable with respect to x, where $c(\cdot)$ and $h(\cdot)$ are given by Definition 1.
2) Both $E_g[c(X)]$ and $E_\theta[c(X)]\ \forall\, \theta \in \Theta$ are finite.
3) θ^* is an interior point of Θ.

Then

$$E_{\theta^*}[c(X)] = E_g[c(X)].$$

Proof: Note that θ^* satisfies the first order necessary condition for optimality. The result follows by using an argument similar to that of [5]. ∎

Given the current observation $\gamma_{N-mK} = \gamma$, Lemma 4.2 states that the optimal parameter θ_i^{k+1} obtained in (7) is the solution to the system of equations

$$\begin{aligned} E_\theta[c(X)] &= E_{g_i^{k+1}}[c(X)] \\ &= (1-\alpha)\frac{E_{\theta_i^k}[I\{H(X)\geq\gamma\}c(X)]}{\mathcal{I}_i(\gamma)} + \alpha E_{\theta_i^k}[c(X)]. \end{aligned}$$

Therefore, given $\Lambda_f^k = \{f_{\theta_1^k}, \ldots, f_{\theta_K^k}\}$, it follows from (10) that an estimate $\widehat{\theta}_i^{k+1}$ of θ_i^{k+1} can be obtained by solving the stochastic counterpart of the above equation

$$\begin{aligned} E_\theta[c(X)] &= (1-\alpha)\frac{\frac{1}{N}\sum_{j=1}^{K}\sum_{x\in\bar{\Lambda}_X^j}\frac{f_{\theta_i^k}(x)}{f_{\theta_j^k}(x)}c(x)}{\hat{\mathcal{I}}_i(\gamma)} \\ &\quad + \alpha E_{\theta_i^k}[c(X)]. \end{aligned}$$

V. Numerical Examples

In this section, we illustrate the performance of PCEDSA on several continuous multi-extremal optimization problems and compare its performance with that of standard CE and a population-based CE referred to as PCE with equal sample allocation (PCEESA). In PCEESA, a population of K distributions is also maintained/updated from generation to generation, and at each iteration of the algorithm, a given sample budget N is equally split into K parts and assigned to each of the K distributions.

The following three benchmark problems are used in our experiments. Function H_1 is a low dimensional problem with a few local optima; however, the maxima are separated by plateaus and are relatively far apart. Both H_2 and H_3 are highly multimodal and the number of local optima increases exponentially with the problem dimension.

(1) Shekel's function ($n = 4$)

$$H_1(x) = -\sum_{i=1}^{5}\Big((x-a_i)^T(x-a_i)+c_i\Big)^{-1},$$

where $a_1 = (4,4,4,4)^T$, $a_2 = (1,1,1,1)^T$, $a_3 = (8,8,8,8)^T$, $a_4 = (6,6,6,6)^T$, $a_5 = (3,7,3,7)^T$, and $c = (0.1, 0.2, 0.2, 0.4, 0.4)$, $x^* \approx (4,4,4,4)^T$, $H_1(x^*) \approx 10.153$.

(2) Trigonometric function ($n = 10$)

$$\begin{aligned} H_2(x) &= -\sum_{i=1}^{n} 8\sin^2\left(7(x_i-0.9)^2\right) \\ &\quad + 6\sin^2\left(14(x_i-0.9)^2\right) + (x_i-0.9)^2 - 1, \end{aligned}$$

where $x^* = (0.9, \ldots, 0.9)^T$, $H_3(x^*) = -1$.

(3) Griewank function ($n = 10$)

$$H_3(x) = -\frac{1}{4000}\sum_{i=1}^{n} x_i^2 + \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) - 1,$$

where $x^* = (0, \ldots, 0)^T$, $H_4(x^*) = 0$.

In our experiments, we have used multivariate normal distributions with independent components. For both PCEDSA and PCEESA, we take population size $K = 5$, and the means μ_i's of the initial population are uniformly selected from $[-40, 40]^n$; $\mathbf{\Sigma}_i$'s are initialized as n-by-n diagonal matrices with all diagonal elements equal to 100. For CE, the initial mean vector is uniformly selected from $[-40, 40]^n$, and covariance matrix is an n-by-n diagonal matrix with all non-zero elements equal to 100. We set $m = 10$ and $\alpha = 0.3$ in PCEDSA, and $L = 10$ for all three algorithms.

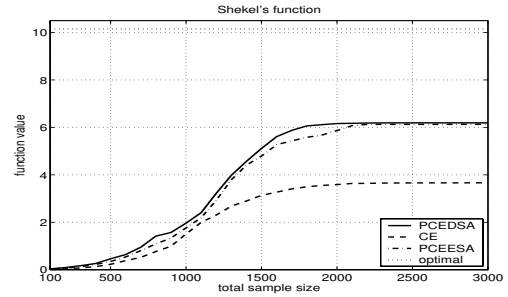
For each test problem, we performed 100 independent replication runs of all three algorithms. Tables I shows the performances of these algorithms when the total number

TABLE I

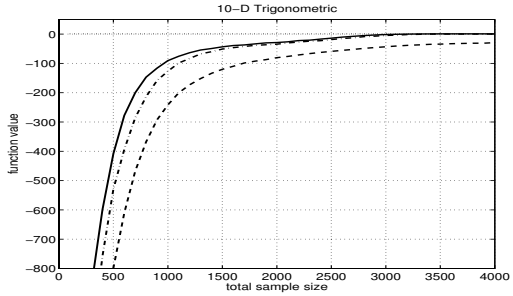
PERFORMANCE OF DIFFERENT ALGORITHMS ON TEST PROBLEMS $H_1 - H_3$, BASED ON 100 INDEPENDENT REPLICATIONS (STANDARD ERRORS IN PARENTHESES).

Test	PCEDSA ($N = 100$)		CE ($N = 100$)		PCEESA ($N = 100$)	
Prob.	$\bar{H}_1^*$	ρ_ε	$\bar{H}_2^*$	ρ_ε	$\bar{H}_3^*$	ρ_ε
H_1	6.20(0.37)	51%	3.66(0.31)	13%	6.12(0.37)	45%
H_2	0.91(0.04)	94%	-29.2(5.11)	19%	0.11(0.18)	78%
H_3	-1.17e-2(1.3e-3)	54%	-7.68e-2(7.3e-3)	5%	-1.23e-2(1.7e-3)	45%

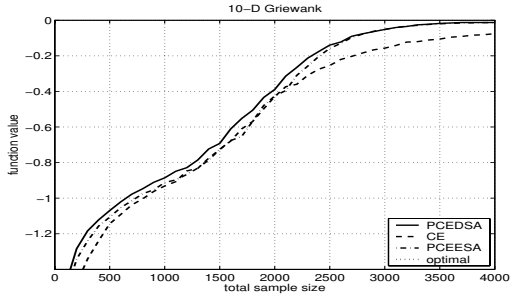
Fig. 2. Average performance of PCEDSA, CE, and PCEESA on test problems H_1 to H_3.



(a) H_1



(b) H_2



(c) H_3

of samples allowed at each iteration is set to $N = 100$. In the table, $\bar{H}_i^*$ indicates the averaged value of function H_i evaluated at the best solution visited by the algorithm, with standard error reported in parenthesis, and ρ_ε is the percentage of ε-optimal solutions ($\varepsilon = 1e{-}3$) obtained out of 100 trials. Here, we call a solution x ε-optimal if it satisfies $|H_i(x^*) - H_i(x)| \leq \varepsilon$. The performance comparison is based on the same amount of computational effort, where the total number of samples allowed for H_1 is set to 3000, and 4000 for H_2 and H_3. In Figure 2, we also plotted the average function values of the current best solution as a function of the number of samples generated.

It is easy to observe that the performance of PCEESA consistently dominates the original CE method, which indicates that a population-based approach may offer more efficiency and robustness in a model-based search method. PCEDSA represents a further enhancement of PCEESA via the use of an intelligent allocation scheme to dynamically allocate computational budgets among different distributions as the sampling process proceeds.

VI. CONCLUSIONS

In this paper, we have proposed a population-based CE method called PCEDSA as a first step to explore the possibility of improving the efficiency and robustness of a typical model based method. Our preliminary numerical results show that PCEDSA offers promising performance. The focus of this paper is on the CE method, however, it could be more useful to view the proposed method as a general population-based framework and apply the dynamic sample allocation idea to other model-based algorithms such as EDAs and MRAS. This avenue of investigation merits further research.

REFERENCES

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volumes 1 and 2.* Athena Scientific, Belmont, MA 1995.

[2] Dorigo, M., L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Trans. on Evolutionary Computation,* **1**, 53-66, 1997.

[3] Fu, M. C., J. Hu, and S. I. Marcus, "Mode-based Randomized Methods for Global Optimization," *Proceedings of the* 17*th International Symposium on Mathematical Theory of Networks and Systems,* Kyoto, Japan, July 2006.

[4] Glover, F., "Tabu Search: A Tutorial," *Interfaces,* **20**, 74-94, 1990.

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

[6] Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* **220**, 671-680, 1983.

[7] Mühlenbein, H., G. Paaß, "From Recombination of Genes to the Estimation of Distributions: *I*. Binary Parameters," *In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, Parallel Problem Solving from Nature - PPSN IV,* pp. 178 − 187, Berlin, Springer Verlag, 1996.

[8] Ross, S., *Stochastic Process, 2rd ed.* John Wiley and Sons, New York, 1995.

[9] Rubinstein, R. Y., D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning.* Springer, New York, 2004.

[10] Shi, L., S. Ólafsson, "Nested Partitions Method for Global Optimization," *Operations Research,* **48**, 390-407, 2000.

[11] Srinivas, M., L. M. Patnaik, "Genetic Algorithms: A Survey," *IEEE Computer,* **27**, 17-26, 1994.

[12] Tang, Z., "Adaptive Partitioned Random Search to Global Optimization," *IEEE Trans. on Automatic Control,* **39**, 2235-2244, 1994.

[13] Zlochin, M., M. Birattari, N. Meuleau, and M. Dorigo, "Model-based Search for Combinatorial Optimization: A Critical Survey," *Annals of Operations Research,* **131**, 373-395, 2004.