Model Reference Adaptive Search: A New Approach to Global Optimization *

Jiaqiao Hu
Department of Electrical and Computer Engineering & Institute for Systems Research
University of Maryland, College Park, MD 20742
JQHU@GLUE.UMD.EDU

Michael C. Fu
Robert H. Smith School of Business & Institute for Systems Research
University of Maryland, College Park, MD 20742
MFU@RHSMITH.UMD.EDU

Steven I. Marcus
Department of Electrical and Computer Engineering & Institute for Systems Research
University of Maryland, College Park, MD 20742
MARCUS@UMD.EDU

ABSTRACT

We present a randomized algorithm called Model Reference Adaptive Search (MRAS) for solving global optimization problems. The algorithm generates at each iteration a group of candidate solutions according to a parameterized probabilistic model. These candidate solutions are then used to update the parameters associated with the probabilistic model in such a way that the future search will be biased toward the region containing high quality solutions. The parameter updating procedure in MRAS is guided by a sequence of implicit reference models that will eventually converge to a model producing only the optimal solutions. We establish global convergence of MRAS in both continuous and combinatorial domains. Numerical studies are also carried out to demonstrate the effectiveness of the algorithm.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization*; G.3 [**Probability and Statistics**]: Probabilistic algorithms; G.4 [**Mathematical Software**]: Algorithm design and analysis

General Terms

Design, algorithms

Keywords

cross-entropy method (CE), estimation of distribution algorithms (EDAs)

1. INTRODUCTION

Global optimization problems arise in a wide range of applications and are often extremely difficult to solve. Following [15], we classify the solution methods for both continuous and combinatorial problems as being either *instance-based* or *model-based*. In *instance-based* methods, searches for new candidate solutions depend directly on previously generated solutions. Some well-known *instance-based* methods are simulated annealing (SA) [7], genetic algorithms (GAs) [14], tabu search [4], and the recently proposed nested partitions (NP) method [13]. In *model-based* algorithms, new solutions are generated via an intermediate probabilistic model that is updated or induced from the previous solutions. The *model-based* search methods are a class of solution techniques introduced fairly recently. In general, most of the algorithms that fall in this category share a similar framework and usually involve the following two phases: (1) Generate candidate solutions (random samples, trajectories) according to a specified probabilistic model (e.g., a parameterized probability distribution on the solution space). (2) Update the parameters associated with the probabilistic model, on the basis of the data collected in the previous step, in order to bias the future search toward "better" solutions. Some well-established *model-based* methods are ant colony optimization (ACO) [3], the cross-entropy (CE) method [12], [2], and the estimation of distribution algorithms (EDAs) [9].

In this paper, we propose a new randomized algorithm, called model reference adaptive search (MRAS), for solving both continuous and combinatorial optimization problems. MRAS resembles CE and EDAs in that it works with a family of parameterized distributions on the solution space. For a detailed discussion of the connection between MRAS and the CE method and additional computational comparison results, the reader is referred to [6]. The motivation behind the method is to use a *sequence* of intermediate *reference* distributions to facilitate and guide the updating of the parameters associated with the family of parameterized distributions during the search process. At each iteration of MRAS, candidate solutions are generated from the distribution (among the prescribed family of distributions) that possesses the minimum Kullback-Leibler (KL)-divergence with respect to the reference model corresponding to the previous iteration. These candidate solutions are in turn used to construct the next distribution that has the minimum

*For a full version of this paper with complete technical developments and detailed proofs, please refer to [6].

KL-divergence with respect to the current reference model, from which future candidate solutions will be generated. We show that for a class of parameterized probability distributions, the so-called Natural Exponential Family (NEF), the algorithm converges to an optimal solution with probability one.

The rest of the paper is organized as follows. In Section 2, we discuss the general ideas behind MRAS. In Section 3, we describe the deterministic version of the algorithm. In Section 4, we describe the Monte Carlo version of MRAS and present its global convergence properties. Illustrative numerical studies on both continuous and combinatorial optimization problems are given in Section 5. Finally, concluding remarks are given in Section 6.

2. BACKGROUND

We consider the following optimization problem:

$$x^* \in \arg\max_{x\in\mathcal{X}} H(x), \quad x \in \mathcal{X} \subseteq \Re^n, \tag{1}$$

where $\mathcal{X}$ is the solution space, and $H(\cdot) : \mathcal{X} \to \Re$ is a deterministic function that is bounded from below, i.e., $\exists\ \mathcal{M} > -\infty$ such that $H(x) \geq \mathcal{M}\ \forall\ x \in \mathcal{X}$. Throughout this paper, we assume that problem (1) has a unique global optimal solution, i.e., there exists $x^* \in \mathcal{X}$ such that $H(x) < H(x^*)$ for all $x \neq x^*,\ x \in \mathcal{X}$. Our primary concern in this paper is on *unconstrained* or *partially constrained* optimization problems. We assume that random sampling can be done easily on $\mathcal{X}$, at least for a class of distributions of interest. For *constrained* optimization problems, we refer the reader to [8] for a discussion of how to convert them to *unconstrained* problems.

To explain the main idea behind MRAS, we consider the following naive *model-based* approach for solving (1). Let $g_0(x) > 0\ \forall x \in \mathcal{X}$ be an initial probability density/mass function (p.d.f./p.m.f.) on the solution space $\mathcal{X}$. At each iteration $k \geq 1$, we compute a new p.d.f. by tilting the old p.d.f. $g_{k-1}(x)$ with the performance function $H(x)$ (for simplicity, here we assume $H(x) > 0\ \forall x \in \mathcal{X}$), i.e.,

$$g_k(x) = \frac{H(x)g_{k-1}(x)}{\int_{\mathcal{X}} H(x)g_{k-1}(dx)}, \quad \forall x \in \mathcal{X}, \tag{2}$$

One direct consequence of this is that each iteration of (2) improves the expected performance. To be precise, let $X = (X_1, \ldots, X_n)$ be a random variable taking values in $\mathcal{X}$. To reduce the notational burden, henceforth X will be used to denote a random variable having the distribution under which the expectation is indicated. Thus, $E_{g_k}[H(X)] = \int_{\mathcal{X}} H(x)g_k(dx)$ and $E_{g_{k-1}}[H(X)] = \int_{\mathcal{X}} H(x)g_{k-1}(dx)$. Then we have

$$\begin{aligned} E_{g_k}[H(X)] &= \frac{E_{g_{k-1}}[(H(X))^2]}{E_{g_{k-1}}[H(X)]} \\ &\geq E_{g_{k-1}}[H(X)]. \end{aligned}$$

Furthermore, it is possible to show that the sequence of p.d.f.'s $\{g_k(\cdot),\ k = 0, 1, \ldots\}$ will converge to a p.d.f. that concentrates only on the set of optimal solutions for arbitrary $g_0(\cdot)$. So we will have $\lim_{k\to\infty} E_{g_k}[H(X)] = H(x^*)$.

However, the above approach is generally of little practical use, due to the following reasons: (i) It is usually not possible to enumerate all the points in the solution space in order to perform the update (2); furthermore, if it were possible, the optimal solution could be immediately identified simply by checking which point has the best performance value. (ii) The p.d.f. $g_k(x)$ constructed at each iteration may not have any structure, and therefore may be very difficult to handle.

To overcome the above difficulties, we consider the Monte Carlo (sampling) version of the above approach and at the same time restrict ourselves to a family of parameterized p.d.f.'s $\{f(\cdot, \theta)\}$, where θ is the parameter vector. In particular, at each iteration k of the algorithm, we look at the projection of $g_k(\cdot)$ on the family of p.d.f.'s $\{f(\cdot, \theta)\}$ and compute the parameter vector θ_k that minimizes the Kullback-Leibler (KL) divergence

$$\mathcal{D}(g_k, f(\cdot, \theta)) := E_{g_k}\left[\ln \frac{g_k(X)}{f(X, \theta)}\right] = \int_{x\in\mathcal{X}} \ln \frac{g_k(x)}{f(x, \theta)} g_k(dx).$$

The benefits of the above consideration are twofold: on the one hand, $f(\cdot, \theta_k)$ often has some special structure and therefore could be much easier to handle than $g_k(\cdot)$. On the other hand, the sequence $\{f(\cdot, \theta_k)\}$ may retain some nice properties of $\{g_k(\cdot)\}$ and converge to a degenerate p.d.f. concentrated on the set of optimal solutions.

First, however, we present the deterministic version of the MRAS algorithm, because it serves as a starting point for deriving the results of Section 4.

3. MRAS – DETERMINISTIC VERSION

Throughout the analysis, we use $P_{\theta_k}(\cdot)$ and $E_{\theta_k}[\cdot]$ to denote the probability and expectation taken with respect to the p.d.f./p.m.f. $f(\cdot, \theta_k)$, and $I_{\{\cdot\}}$ to denote the indicator function, i.e.,

$$I_{\{A\}} := \begin{cases} 1 & \text{if event } A \text{ holds,} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, under our notational convention,

$$P_{\theta_k}\left(H(X) \geq \gamma\right) = \int_{x\in\mathcal{X}} I_{\{H(x)\geq\gamma\}} f(dx, \theta_k), \text{ and}$$

$$E_{\theta_k}[H(X)] = \int_{x\in\mathcal{X}} H(x) f(dx, \theta_k).$$

3.1 Algorithm Description

The MRAS_0 algorithm requires specification of a parameter ρ, which determines the approximate proportion of samples that will be used to update the probabilistic model. At successive iterations of the algorithm, a sequence $\{\gamma_k, k = 1, 2, \ldots\}$, i.e., the $(1-\rho)$-quantiles with respect to the sequence of p.d.f's $\{f(\cdot, \theta_k)\}$, are calculated at step 1 of MRAS_0. These quantile values are then used in step 2 to construct a sequence of non-decreasing thresholds $\{\bar{\gamma}_k, k = 1, 2, \ldots\}$; and only those candidate solutions that have performances better than these thresholds will be used in parameter updating (cf. equation (3)). As we will see, the theoretical convergence of MRAS_0 is unaffected by the value of the parameter ρ. The purpose of ρ in our approach is to concentrate the computational effort on the set of elite/promising samples, which is a standard technique employed in most of the population-based approaches, like GAs and EDAs.

Algorithm MRAS$_0$: deterministic version

- **Initialization:** Specify the parameter $\rho \in (0,1]$, a small number $\varepsilon \geq 0$, a continuous and strictly increasing function $S(\cdot): \Re \to \Re^+$, and an initial p.d.f./p.m.f. $f(x,\theta_0) > 0\ \forall x \in \mathcal{X}$. Set the iteration counter $k = 0$.
- **Repeat until a specified stopping rule is satisfied:**
  1. Calculate the $(1-\rho)$-quantile
  $$\gamma_{k+1} := \sup_l \left\{ l : P_{\theta_k}(H(X) \geq l) \geq \rho \right\}.$$
  2. **if** $k = 0$, **then** set $\bar{\gamma}_{k+1} = \gamma_{k+1}$.
  elseif $k \geq 1$
  if $\gamma_{k+1} \geq \bar{\gamma}_k + \varepsilon$, **then** set $\bar{\gamma}_{k+1} = \gamma_{k+1}$.
  else set $\bar{\gamma}_{k+1} = \bar{\gamma}_k$.
  endif
  endif
  3. Compute the new parameter vector θ_{k+1} as
  $$\theta_{k+1} = \arg\max_{\theta\in\Theta} E_{\theta_k}\left[\frac{[S(H(X))]^k}{f(X,\theta_k)} I_{\{H(X)\geq\bar{\gamma}_{k+1}\}} \ln f(X,\theta)\right]. \quad (3)$$
  4. Set $k = k+1$.

During the initialization step of MRAS$_0$, a small number ε and a continuous and strictly increasing function $S(\cdot): \Re \to \Re^+$ are also specified. The function $S(\cdot)$ is used to account for the cases where the values of $H(x)$ are negative for some x, and the parameter ε ensures that each strict increment in the sequence $\{\bar{\gamma}_k\}$ is lower bounded, i.e.,

$$\inf_{\substack{\bar{\gamma}_{k+1}\neq\bar{\gamma}_k \\ k=1,2,\ldots}} (\bar{\gamma}_{k+1} - \bar{\gamma}_k) \geq \varepsilon.$$

We require ε to be strictly positive for continuous problems, and non-negative for discrete problems.

The following lemma shows that there is a sequence of reference models $\{g_k(\cdot),\ k = 1,2,\ldots\}$ implicit in MRAS$_0$, and the parameter θ_{k+1} computed at step 3 indeed minimizes the KL-divergence $\mathcal{D}(g_{k+1}, f(\cdot,\theta))$.

LEMMA 3.1. *The parameter θ_{k+1} computed at the kth iteration of the MRAS$_0$ algorithm minimizes the KL-divergence $\mathcal{D}(g_{k+1}, f(\cdot,\theta))$, where*

$$g_{k+1}(x) := \frac{S(H(x))I_{\{H(x)\geq\bar{\gamma}_{k+1}\}} g_k(x)}{E_{g_k}\left[S(H(X))I_{\{H(X)\geq\bar{\gamma}_{k+1}\}}\right]} \quad \forall x \in \mathcal{X},\quad k = 1,\ldots,$$

and $g_1(x) := \dfrac{I_{\{H(x)\geq\bar{\gamma}_1\}}}{E_{\theta_0}\left[\frac{I_{\{H(X)\geq\bar{\gamma}_1\}}}{f(X,\theta_0)}\right]}.$

3.2 Global Convergence

Throughout this paper, we restrict our discussions to a particular family of p.d.f.'s/p.m.f.'s called the natural exponential family (NEF), for which the global convergence properties can be established. We start by stating the definition of NEF and some regularity conditions.

DEFINITION 3.1. *A parameterized family of p.d.f's $\{f(\cdot,\theta),\ \theta \in \Theta \subseteq \Re^m\}$ on $\mathcal{X}$ is said to belong to the natural exponential family (NEF) if there exist functions $h(\cdot): \Re^n \to \Re$, $\Gamma(\cdot): \Re^n \to \Re^m$, and $K(\cdot): \Re^m \to \Re$ such that*

$$f(x,\theta) = \exp\left\{\theta^T\Gamma(x) - K(\theta)\right\} h(x), \quad \forall\theta \in \Theta, \quad (4)$$

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

Many common p.d.f.'s/p.m.f.'s belong to the NEF, e.g., Gaussian, Poisson, binomial, geometric, and certain multivariate forms of them.

Assumptions:

A1. *There exists a compact set $\Pi \subseteq \mathcal{X}$ such that the level set $\{x : H(x) \geq \bar{\gamma}_1\} \subseteq \Pi$, where $\bar{\gamma}_1 = \sup_l\{l : P_{\theta_0}(H(X) \geq l) \geq \rho\}$ is defined as in the MRAS$_0$ algorithm.*

A2. *For any given constant $\xi < H(x^*)$, the set $\{x : H(x) \geq \xi\}$ has a strictly positive Lebesgue measure.*

A3. *For any given constant $\delta > 0$, $\sup_{x\in A_\delta} H(x) < H(x^*)$, where $A_\delta := \{x : \|x - x^*\| \geq \delta\}$.*

A4. *The maximizer of equation (3) is an interior point of Θ for all k.*

A5. *$\sup_{\theta\in\Theta} \|\exp\{\theta^T\Gamma(x)\}\Gamma(x)h(x)\|$ is integrable/summable with respect to x, where θ, $\Gamma(\cdot)$, and $h(\cdot)$ are defined as in Definition 3.1.*

A6. *$\Gamma(\cdot): \Re^m \to \Re^n$ given in Definition 3.1 is a continuous mapping.*

We have the following convergence result for the MRAS$_0$ algorithm.

THEOREM 3.1. **(Continuous Optimization)** *Let $\{\theta_k,\ k = 1,2,\ldots\}$ be the sequence of parameters generated by MRAS$_0$. If $\varepsilon > 0$ and assumptions A1$-$A6 are satisfied, then*

$$\lim_{k\to\infty} E_{\theta_k}\left[\Gamma(X)\right] = \Gamma(x^*). \quad (5)$$

The convergence result in Theorem 3.1 is much stronger than it appears to be. For example, when $\Gamma(x)$ is a one-to-one function (which is the case for many NEFs encountered in practice), the convergence result (5) can be equivalently written as $\Gamma^{-1}\left(\lim_{k\to\infty} E_{\theta_k}\left[\Gamma(X)\right]\right) = x^*$. Also note that the limit in equation (5) is component-wise. For some particular p.d.f.'s/p.m.f.'s, the solution vector x itself will be a component of $\Gamma(x)$ (e.g., multivariate normal distribution). Under these circumstances, we can disregard the redundant components and interpret equation (5) as $\lim_{k\to\infty} E_{\theta_k}[X] = x^*$. Another special case of particular interest is when the components of the random vector $X = (X_1,\ldots,X_n)$ are independent, i.e., each has a univariate p.d.f. of the form

$$f(x_i,\vartheta_i) = \exp(x_i\vartheta_i - K(\vartheta_i))h(x_i),\ \vartheta_i \in \Re, \forall\ i = 1,\ldots,n.$$

In this case, since the p.d.f. of the random vector X is simply the product of the marginal p.d.f.'s, we will clearly have $\Gamma(x) = x$. Thus, equation (5) is again equivalent to $\lim_{k\to\infty} E_{\theta_k}[X] = x^*$, where $\theta_k := (\vartheta_1^k,\ldots,\vartheta_n^k)$, and ϑ_i^k is the value of ϑ_i at the kth iteration. Some of these special cases are addressed below.

COROLLARY 3.1. **(Multivariate Normal)** *If multivariate normal p.d.f.'s are used in MRAS$_0$, i.e.,*

$$f(x,\theta_k) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right), \tag{6}$$

where $\theta_k := (\mu_k;\ \Sigma_k)$, $\varepsilon > 0$, and assumptions A1−A4 are satisfied, then

$$\lim_{k\to\infty} \mu_k = x^*, \quad \textit{and} \quad \lim_{k\to\infty} \Sigma_k = 0_{n\times n},$$

where $0_{n\times n}$ represents an n-by-n zero matrix.

COROLLARY 3.2. **(Independent Univariate)** *If the components of the random vector $X = (X_1,\ldots,X_n)$ are independent, each has a univariate p.d.f. of the form*

$$f(x_i,\vartheta_i) = \exp(x_i\vartheta_i - K(\vartheta_i))h(x_i),\ \vartheta_i \in \Re, \forall\ i = 1,\ldots,n,$$

$\varepsilon > 0$, and assumptions A1−A6 are satisfied, then

$$\lim_{k\to\infty} E_{\theta_k}[X] = x^*, \quad \textit{where}\ \ \theta_k := (\vartheta_1^k,\ldots,\vartheta_n^k).$$

Note that the convergence of MRAS$_0$ for discrete optimization problems with *infinite countable* domains can be shown similarly by following the proof of Theorem 3.1 in [6], provided that all assumptions A1−A6 except A2 are satisfied. It is worth mentioning that for problems with *finite* solution spaces, assumptions A1 and A3 are automatically satisfied. Furthermore, the input parameter ε need not be strictly positive as in the continuous case.

THEOREM 3.2. **(Discrete Optimization with Finite Domain)** *If the solution space $\mathcal{X}$ is finite and assumptions A4 and A5 are satisfied, then*

$$\lim_{k\to\infty} E_{\theta_k}[\Gamma(X)] = \Gamma(x^*).$$

4. MRAS – MONTE CARLO VERSION

The MRAS$_0$ algorithm describes the idealized situation where quantile values and expectations can be evaluated exactly. In practice, we will usually resort to its stochastic counterpart, where only a finite number of samples are used and expected values are replaced with their corresponding sample averages. However, the theoretical convergence can no longer be guaranteed for a simple stochastic counterpart of MRAS$_0$. We can only expect the algorithm to converge if the expected values in the MRAS$_0$ algorithm are closely approximated. Obviously, the quality of the approximation will depend on the number of samples to be used in the simulation, but it is difficult to determine in advance the appropriate number of samples. A sample size too small will cause the algorithm to fail to converge and result in poor quality solutions, whereas a sample size too large may lead to high computational cost.

As mentioned earlier, the parameter ρ, to some extent, will affect the performance of the algorithm. Large values of ρ mean that almost all samples generated, whether "good" or "bad", will be used to update the probabilistic model, which could slow down the convergence process. On the other hand, since a good estimate will necessarily require a reasonable amount of valid samples, the quantity ρN (i.e., the approximate amount of samples that will be used in parameter updating) cannot be too small. Thus, small values of ρ will require a large number of samples to be generated at each iteration and may result in significant simulation efforts. Thus, for a given problem, to determine a priori which ρ gives a satisfactory performance may be difficult.

In order to address the above difficulties, we adopt the same idea as in [5] and propose a modified Monte Carlo version of MRAS$_0$ in which the sample size N is adaptively increasing and the parameter ρ is adaptively decreasing.

4.1 Algorithm Description

Algorithm MRAS$_1$: Monte Carlo version

- **Initialization:** Specify $\rho_0 \in (0,1]$, initial sample size $N_0 > 1$, $\varepsilon \geq 0$, $\alpha > 1$, mixing coefficient $\lambda \in (0,1]$, a function $S(\cdot): \Re \to \Re^+$, and an initial p.d.f. $f(x,\theta_0) > 0\ \forall x \in \mathcal{X}$. Set $\widetilde{\theta}_0 \leftarrow \theta_0$, $k \leftarrow 0$.
- **Repeat until a specified stopping rule is satisfied:**
  1. Generate N_k i.i.d. samples $X_1^k,\ldots,X_{N_k}^k$ according to $\widetilde{f}(\cdot,\widetilde{\theta}_k) := (1-\lambda)f(\cdot,\widetilde{\theta}_k) + \lambda f(\cdot,\theta_0)$.
  2. Compute the sample $(1-\rho_k)$-quantile $\widetilde{\gamma}_{k+1}(\rho_k,N_k) := H_{(\lceil(1-\rho_k)N_k\rceil)}$, where $\lceil a\rceil$ is the smallest integer greater than a, and $H_{(i)}$ is the ith order statistic of the sequence $\left\{H(X_i^k),\ i = 1,\ldots,N_k\right\}$.
  3. **If** $k = 0$ **or** $\widetilde{\gamma}_{k+1}(\rho_k,N_k) \geq \bar{\gamma}_k + \frac{\varepsilon}{2}$, **then**

     3a. Set $\bar{\gamma}_{k+1} \leftarrow \widetilde{\gamma}_{k+1}(\rho_k,N_k)$, $\rho_{k+1} \leftarrow \rho_k$, $N_{k+1} \leftarrow N_k$.

     else, find the largest $\bar{\rho} \in (0,\rho_k)$ such that $\widetilde{\gamma}_{k+1}(\bar{\rho},N_k) \geq \bar{\gamma}_k + \frac{\varepsilon}{2}$.

     3b. **If** such a $\bar{\rho}$ exists, **then** set $\bar{\gamma}_{k+1} \leftarrow \widetilde{\gamma}_{k+1}(\bar{\rho},N_k)$, $\rho_{k+1} \leftarrow \bar{\rho}$, $N_{k+1} \leftarrow N_k$.

     3c. **else** (if no such $\bar{\rho}$ exists), set $\bar{\gamma}_{k+1} \leftarrow \bar{\gamma}_k$, $\rho_{k+1} \leftarrow \rho_k$, $N_{k+1} \leftarrow \lceil\alpha N_k\rceil$.

     endif
  4. Compute $\widetilde{\theta}_{k+1}$ as

$$\arg\max_{\theta\in\Theta} \frac{1}{N_k}\sum_{i=1}^{N_k} \frac{[S(H(X_i^k))]^k}{\widetilde{f}(X_i^k,\widetilde{\theta}_k)} I_{\left\{H(X_i^k)\geq\bar{\gamma}_{k+1}\right\}} \ln f(X_i^k,\theta). \tag{7}$$

  5. Set $k \leftarrow k+1$.

Roughly speaking, the MRAS$_1$ algorithm is essentially a Monte Carlo version of MRAS$_0$ except that the parameter ρ and the sample size N may change from one iteration to another. The rate of increase in the sample size is controlled by an extra parameter $\alpha > 1$, specified during the initialization step. For example, if the initial sample size is N_0, then after k increases, the sample size will be approximately $\lceil\alpha^k N_0\rceil$.

At each iteration k, random samples are drawn from the density/mass function $\widetilde{f}(\cdot,\widetilde{\theta}_k)$, which is a mixture of the initial density $f(\cdot,\theta_0)$ and the density calculated from the previous iteration $f(\cdot,\widetilde{\theta}_k)$ (cf. e.g., [1] for a similar idea in the context of multiarmed bandit models). Intuitively, mixing in the initial density forces the algorithm to explore the entire solution space and to maintain a global perspective during the search process. Also note that if $\lambda = 1$, then random samples will always be drawn from the initial density,

in which case, MRAS$_1$ becomes a pure random sampling method.

At step 2, the sample $(1-\rho_k)$-quantile $\widetilde{\gamma}_{k+1}$ is calculated by first ordering the sample performances $H(X_i^k)$, $i = 1, \ldots, N_k$ from smallest to largest, $H_{(1)} \leq H_{(2)} \leq \cdots \leq H_{(N_k)}$, and then taking the $\lceil (1-\rho_k)N_k \rceil$th order statistic. We use the function $\widetilde{\gamma}_{k+1}(\rho_k, N_k)$ to emphasize the dependencies of $\widetilde{\gamma}_{k+1}$ on both ρ_k and N_k, so that different sample quantile values used during one iteration can be distinguished by their arguments.

Step 3 of MRAS$_1$ is used to extract a sequence of non-decreasing thresholds $\{\bar{\gamma}_k, k = 1, 2 \ldots\}$ from the sequence of sample quantiles $\{\widetilde{\gamma}_k\}$, and to determine the appropriate values of ρ_{k+1} and N_{k+1} to be used in subsequent iterations. This step is carried out as follows. At each iteration k, we first check whether the inequality $\widetilde{\gamma}_{k+1}(\rho_k, N_k) \geq \bar{\gamma}_k + \frac{\varepsilon}{2}$ is satisfied, where $\bar{\gamma}_k$ is the threshold value used in the previous iteration. If the inequality holds, then it means that both the current ρ_k value and the current sample size N_k are satisfactory; thus we proceed to step 3*a* and update the parameter vector $\widetilde{\theta}_{k+1}$ in step 4 by using $\widetilde{\gamma}_{k+1}(\rho_k, N_k)$. Otherwise, it indicates that either ρ_k is too large or the sample size N_k is too small. To determine which, we fix the sample size N_k and check if there exists a smaller $\bar{\rho} < \rho_k$ such that the above inequality can be satisfied with the new sample $(1-\bar{\rho})$-quantile. If such a $\bar{\rho}$ does exist, then the current sample size N_k is still deemed acceptable, and we only need to decrease the ρ_k value. Accordingly, the parameter vector is updated in step 4 by using the sample $(1-\bar{\rho})$-quantile. On the other hand, if no such $\bar{\rho}$ can be found, then the parameter vector is updated by using the threshold $\bar{\gamma}_k$ calculated during the previous iteration and the sample size N_k is increased by a factor α.

We make the following assumption about the parameter vector $\widetilde{\theta}_{k+1}$ computed at step 4:

Assumption A4′. The parameter vector $\widetilde{\theta}_{k+1}$ computed at step 4 of MRAS$_1$ is an interior point of Θ for all k.

It is important to note that the set $\{x : H(x) \geq \bar{\gamma}_{k+1}, x \in \{X_1^k, \ldots, X_{N_k}^k\}\}$ could be empty if step 3*c* is visited. If this happens, the right hand side of equation (7) will be equal to zero, so any $\theta \in \Theta$ is a maximizer, and we define $\widetilde{\theta}_{k+1} := \widetilde{\theta}_k$ in this case.

4.2 Global Convergence

In this section, we present the convergence properties of the MRAS$_1$ algorithm for natural exponential families (NEFs). We denote by $P_{\widetilde{\theta}_k}(\cdot)$ and $E_{\widetilde{\theta}_k}[\cdot]$ the respective probability and expectation taken with respect to the p.d.f. $f(\cdot, \widetilde{\theta}_k)$, and $\widetilde{P}_{\widetilde{\theta}_k}(\cdot)$ and $\widetilde{E}_{\widetilde{\theta}_k}[\cdot]$ the respective probability and expectation taken with respect to $\widetilde{f}(\cdot, \widetilde{\theta}_k)$. Note that since the sequence $\{\widetilde{\theta}_k\}$ results from random samples generated at each iteration of MRAS$_1$, these quantities are also random.

To present the main theorem, we require one more assumption.

Assumption B1. *There exists a compact set Π_ε such that $\{x : H(x) \geq H(x^*) - \varepsilon\} \subseteq \Pi_\varepsilon$. Moreover, $f(x, \theta_0)$ is bounded away from zero on Π_ε, i.e., $f_* := \inf_{x \in \Pi_\varepsilon} f(x, \theta_0) > 0$.*

THEOREM 4.1. **(Continuous Optimization)** *Let $\varepsilon > 0$, and define the ε-optimal set $\mathcal{O}_\varepsilon := \{x : H(x) \geq H(x^*) - \varepsilon\}$. If assumptions A2, A4′, A5, and B1 are satisfied, then there exists a random variable $\mathcal{K}$ such that, w.p.1., $\mathcal{K} > 0$ and*
1. $\bar{\gamma}_k > H(x^*) - \varepsilon, \quad \forall\ k \geq \mathcal{K}$
2. $E_{\widetilde{\theta}_{k+1}}[\Gamma(X)] \in CONV\{\Gamma(\mathcal{O}_\varepsilon)\}, \quad \forall\ k \geq \mathcal{K},$
where $CONV\{\Gamma(\mathcal{O}_\varepsilon)\}$ indicates the convex hull of the set $\Gamma(\mathcal{O}_\varepsilon)$.
Furthermore, let β be a positive constant satisfying the condition that the set $\{x : S(H(x)) \geq \frac{1}{\beta}\}$ has a strictly positive Lebesgue measure. If assumptions A2, A3, A4′, A5, A6, and B1 are satisfied and $\alpha > (\beta S^*)^2$, where $S^* := S(H(x^*))$, then
3. $\lim_{k\to\infty} E_{\widetilde{\theta}_k}[\Gamma(X)] = \Gamma(x^*)$ w.p.1.

The following results are now immediate.

COROLLARY 4.1. **(Multivariate Normal)** *If multivariate normal p.d.f.'s are used in MRAS$_1$, i.e.,*

$$f(x, \widetilde{\theta}_k) = \frac{1}{\sqrt{(2\pi)^n |\widetilde{\Sigma}_k|}} \exp\left(-\frac{1}{2}(x - \widetilde{\mu}_k)^T \widetilde{\Sigma}_k^{-1}(x - \widetilde{\mu}_k)\right),$$

$\varepsilon > 0$, $\alpha > (\beta S^*)^2$, and assumptions A2, A3 and A4′ are satisfied, then

$$\lim_{k\to\infty} \widetilde{\mu}_k = x^*, \quad \text{and} \quad \lim_{k\to\infty} \widetilde{\Sigma}_k = 0_{n\times n} \quad w.p.1.$$

COROLLARY 4.2. **(Independent Univariate)** *If the components of the random vector $X = (X_1, X_2, \ldots, X_n)$ are independent, each with a univariate p.d.f. of the form*

$$f(x_i, \vartheta_i) = \exp(x_i\vartheta_i - K(\vartheta_i))h(x_i), \ \vartheta_i \in \Re, \forall\ i = 1, \ldots, n,$$

$\varepsilon > 0$, $\alpha > (\beta S^*)^2$, and assumptions A2, A3, A4′, A5, and B1 are satisfied, then

$$\lim_{k\to\infty} E_{\widetilde{\theta}_k}[X] = x^* \quad w.p.1, \quad \text{where} \quad \widetilde{\theta}_k := (\vartheta_1^k, \ldots, \vartheta_n^k).$$

THEOREM 4.2. **(Discrete Optimization with Finite Domains)** *If the solution space $\mathcal{X}$ is finite, $\alpha > 1$, and assumptions A4′, and A5 are satisfied, then*

$$\lim_{k\to\infty} E_{\widetilde{\theta}_k}[\Gamma(X)] = \Gamma(x^*) \quad w.p.1.$$

5. NUMERICAL EXAMPLES

In this section, we illustrate the performance of the MRAS method for both continuous and combinatorial optimization problems. In the former case, we test the algorithm on various functions that are well-known in global optimization. In the latter case, we apply the algorithm to several Asymmetric Traveling Salesman Problems (ATSP), which are typical representatives of NP-hard combinatorial optimization problems.

We now discuss some implementation issues of the MRAS$_1$ algorithm.

1. Since all examples considered are minimization problems, the function $S(\cdot)$ is initialized as a strictly decreasing function. Throughout this section, we take $S(H(x)) := \exp\{-rH(x)\}$, where r is a positive constant.

2. In actual implementation of the algorithm, a smoothed parameter updating procedure as in [10] is used, i.e., first a smoothed parameter vector $\widehat{\theta}_{k+1}$ is computed at each iteration k according to

   $$\widehat{\theta}_{k+1} := \upsilon\,\widetilde{\theta}_{k+1} + (1-\upsilon)\widehat{\theta}_k,\ \forall\, k = 0, 1, \ldots, \text{ and } \widehat{\theta}_0 := \widetilde{\theta}_0,$$

   where $\widetilde{\theta}_{k+1}$ is the parameter vector computed at step 4 of MRAS$_1$, and $\upsilon \in (0, 1]$ is the smoothing parameter; then $f(x, \widehat{\theta}_{k+1})$ (instead of $f(x, \widetilde{\theta}_{k+1})$) is used in step 1 to generate new samples.

3. In the numerical experiments, we stop the algorithm when either one of the following two conditions is satisfied at iteration k:

   (1) $\max_{1\le i\le d} |\bar{\gamma}_k - \bar{\gamma}_{k+i}| \le \tau$;
   (2) $N_k > N_{\max}$;

   where $\tau > 0$ is a predefined tolerance level, d is a positive integer, and $N_{\max}$ is the maximum number of samples allowed per iteration.

5.1 Continuous Optimization

In our preliminary experiments, we take the family of parameterized p.d.f.'s to be multivariate normal p.d.f.'s. A mean vector μ_0 and a covariance matrix Σ_0 are specified initially. By Corollary 4.1, the sequence of mean vectors $\{\widetilde{\mu}_k\}$ generated by the algorithm will converge to the optimal solution x^*, and the sequence of covariance matrices $\{\widetilde{\Sigma}_k\}$ to the zero matrix.

The following five functions $\{H_i,\ i = 1, \ldots, 5\}$ are used to test the algorithm.

(1) Quadratic function

$$H_1(x) = \sum_{i=1}^{3} x_i^2, \quad \text{where } x = (x_1, x_2, x_3).$$

The function has a global minimum $f(0, 0, 0) = 0$.

(2) Two-dimensional Rosenbrock function

$$H_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1^2), \quad \text{where } x = (x_1, x_2).$$

The function has a global minimum $f(1, 1) = 0$.

(3) Shekel's Foxholes

$$H_3(x) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{j,i})^6}},$$

where $a_{j,1} = \{-32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32\}$, $a_{j,2} = \{-32, -32, -32, -32, -32, -16, -16, -16, -16, -16, 0, 0, 0, 0, 0, 16, 16, 16, 16, 16, 32, 32, 32, 32, 32\}$, and $x = (x_1, x_2)$. The function has 24 local minima and one global minimum at $f(-32, -32) \approx 0.998004$.

(4) Corana's Parabola

$$H_4(x) = \sum_{i=1}^{4} \begin{cases} 0.15[0.05\,\mathrm{sgn}(z_i) - z_i]^2 d_j & \text{if } |x_i - z_i| < 0.05, \\ d_i x_i^2 & \text{otherwise,} \end{cases}$$

where

$$z_i = 0.2\left\lfloor \left| \frac{x_i}{2} \right| + 0.49999 \right\rfloor \mathrm{sgn}(x_i),$$

$d = \{1, 1000, 10, 100\}$, and $x = (x_1, x_2, x_3, x_4)$. It has a global minimum $f(0, 0, 0, 0) = 0$.

(5) Goldstein-Price function

$$H_5(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))$$
$$(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)),$$

where $x = (x_1, x_2)^T$. The function has four local minima and a global minimum $f(0, -1) = 3$.

For all five problems, the same set of parameters is used to test MRAS: $\varepsilon = 10^{-5}$, initial sample size $N_0 = 100$, $\rho_0 = 0.2$, $\lambda = 0.02$, $\alpha = 1.5$, $r = 0.1$, the stopping control parameters $d = 5$, $\tau = 10^{-5}$, $N_{max} = 50000$, and the smoothing parameter $\upsilon = 0.5$. The initial mean vector μ_0 is a d-by-1 vector of all $10s$, and Σ_0 is a d-by-d diagonal matrix with all diagonal elements equal to 200, where d is the dimension of the problem. Table 1 shows the performance of the algorithm

H_i	N_{total} (std)	$\widetilde{H}_i^*$ (std)	$H_i(x^*)$	M_ε
H_1	4.4e+03(6.8e+01)	9.9e-09(1.1e-09)	0	50
H_2	1.2e+04(4.9e+02)	2.3e-09(3.1e-10)	0	50
H_3	2.2e+04(7.2e+02)	2.40(4.15e-01)	0.998	37
H_4	7.4e+03(1.6e+02)	0.00(0.00e-00)	0	50
H_5	5.8e+03(1.4e+02)	3.00(5.30e-10)	3	50

Table 1: Performance of MRAS on five test functions, based on 50 independent simulation runs.

on the five test functions. For each function, we performed 50 independent simulation runs of the algorithm, and the means and standard errors are reported in the table, where N_{total} is the total number of function evaluations, and $\widetilde{H}_i^*$ is the value of the function $H_i(\cdot)$ at the final value of the estimated optimum. The optimal value $H_i(x^*)$ is included for reference, and M_ε indicates the number of runs that an ε-optimal solution was found out of 50 trials. The algorithm performs quite well in most cases, except for H_3, where only 37 ε-optimal solutions were found. H_3 represents a class of continuous optimization problems that are extremely difficult to solve for most *model-based* sampling approaches. A graphical representation of the function H_3 is given in Figure 1. Notice that the function values at the 25 "holes" (local minima) are very close to each other; thus in order to locate the global optimal solution, the algorithm must make sure that samples are drawn from the right "hole", and there must be enough samples to fall in this "hole" to guarantee that the parameter vectors are updated in the right direction.

Table 2 gives the performance of MRAS on function H_3 using different sample sizes and ρ values (all other parameters are the same as before). We see that for the $N_0 = 200$ cases,

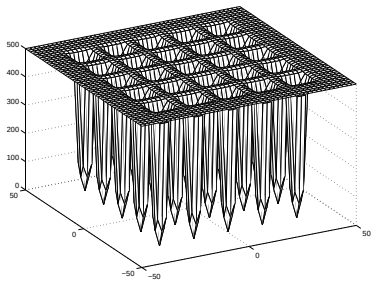

Figure 1: Shekel's Foxholes, $-50 \le x_i \le 50,\ i = 1, 2.$

ε-optimal solutions were found in more than 90% of the total simulation runs; whereas for the $N_0 \ge 500$ cases, ε-optimal solutions were found in all 50 runs.

parameters	N_{total} (std)	$\widetilde{H}_3^*$ (std)	M_ε
N_0=200, ρ_0=0.2	2.3e+4(6.8e+2)	1.14(0.06)	45
N_0=200, ρ_0=0.1	2.2e+4(7.1e+2)	1.08(0.05)	47
N_0=500, ρ_0=0.2	3.0e+4(6.7e+2)	0.998(3.4e-11)	50
N_0=500, ρ_0=0.1	2.8e+4(8.7e+2)	0.998(3.9e-11)	50
N_0=1000, ρ_0=0.2	5.6e+4(8.3e+2)	0.998(3.4e-11)	50
N_0=1000, ρ_0=0.1	4.3e+4(8.5e+2)	0.998(3.8e-11)	50

Table 2: Performance of MRAS on test function H_3**. The optimum** $H_3(x^*) \approx 0.998004$**.**

We also applied MRAS to a 10-dimensional trigonometric function and a 10-dimensional Rosenbrock function.

(6) Trigonometric function

$$H_6(x) = \sum_{i=1}^{10} 8\sin^2\left(7(x_i - 0.9)^2\right) + 6\sin^2\left(14(x_i - 0.9)^2\right) + (x_i - 0.9)^2.$$

The function has a global minimum at $x^* = (0.9, \ldots, 0.9)^T$, and $H_6(x^*) = 0$.

(7) Rosenbrock function

$$H_7(x) = \sum_{i=1}^{9} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$$

The function has a global minimum at $x^* = (1, \ldots, 1)^T$, and $H_7(x^*) = 0$.

In our preliminary experiments with function H_6, we use the same set of parameters as before. Test results in Table 3 indicate that the algorithm may frequently get trapped in local optimal solutions. We believe this is due to the fact that the covariance matrix often converges too quickly to a zero matrix.

In order to reduce the convergence speed of the covariance matrix, we reran the algorithm with smaller values of r and υ. Table 4 shows the performance of the algorithm with $r = 0.01$ and $\upsilon = 0.2$. We see that the algorithm successfully escaped the local optima in all simulation runs. However, decreasing the values of r and υ also slows down the convergence of the algorithm; note that the total number of function evaluations in all cases almost increased by a factor of 10 as compared to Table 3. For a given problem, how to choose the most appropriate parameters in MRAS is still an open issue.

parameters	N_{total} (std)	$\widetilde{H}_6^*$ (std)	M_ε
N_0=200, ρ_0=0.1	5.6e+4(4.4e+3)	1.05(0.17)	20
N_0=200, ρ_0=0.2	4.7e+4(2.4e+3)	1.02(0.16)	16
N_0=500, ρ_0=0.1	7.8e+4(3.9e+3)	0.29(0.06)	28
N_0=500, ρ_0=0.2	6.2e+4(3.0e+3)	0.64(0.11)	23

Table 3: Performance of MRAS ($r = 0.1,\ \upsilon = 0.5$**) on test function** H_6**, based on** 50 **independent runs.**

parameters	N_{total} (std)	$\widetilde{H}_6^*$ (std)	M_ε
N_0=200, ρ_0=0.1	5.8e+5(4.6e+4)	3.3e-7(2.6e-8)	50
N_0=200, ρ_0=0.2	4.2e+5(4.6e+4)	3.6e-7(2.6e-8)	50
N_0=500, ρ_0=0.1	6.0e+5(4.8e+4)	2.9e-7(2.3e-8)	50
N_0=500, ρ_0=0.2	5.4e+5(3.1e+4)	2.6e-7(1.9e-8)	50

Table 4: Performance of MRAS ($r = 0.01,\ \upsilon = 0.2$**) on test function** H_6**.**

The same set parameters are also used to solve the function H_7, and the numerical results are reported in Table 5.

parameters	N_{total} (std)	$\widetilde{H}_7^*$ (std)	M_ε
N_0=200, ρ_0=0.1	2.7e+5(1.3e+4)	2.4e-8(2.9e-9)	50
N_0=200, ρ_0=0.2	2.6e+5(1.5e+4)	2.3e-8(2.2e-9)	50
N_0=500, ρ_0=0.1	3.3e+5(1.4e+4)	1.8e-8(1.8e-9)	50
N_0=500, ρ_0=0.2	3.6e+5(1.6e+4)	2.9e-8(3.4e-9)	50

Table 5: Performance of MRAS ($r = 0.01,\ \upsilon = 0.2$**) on test function** H_7**.**

5.2 Combinatorial Optimization

In this section, we present the performance of MRAS on various ATSP problems. All test cases are taken from the URL http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95.

For each ATSP problem with N_c cities, an N_c-by-N_c distance matrix G is given, whose (i, j)th element $G_{i,j}$ represents the distance from city i to city j. The goal is to find the shortest path that visits all the cities and returns to the starting city.

We use the same technique as in [11] for solving these problems, i.e., we associate with each distance matrix G an initial state transition matrix $\widetilde{P}_0$, whose $(i, j)th$ element specifies the probability of transitioning from city i to city j. Thus, at each iteration of MRAS the following two steps are fundamental: (1) Generating random (admissible) tours according to the transition matrix and evaluating the performance of each sample tour. (2) Updating the transition matrix based on the sample tours generated from the previous step.

The performance of the algorithm on various ATSP problems is reported in Table 6. For each of the 7 instances,

ATSP	N_c	N_{total} (std err)	H_{best}	H_*	H^*	δ_*	δ^*	δ (std err)
ftv33	34	7.95e+04(3.25e+03)	1286	1364	1286	0.061	0.000	0.023(0.008)
ftv35	36	1.02e+05(3.08e+03)	1473	1500	1475	0.018	0.001	0.008(0.002)
ftv38	39	1.31e+05(4.90e+03)	1530	1563	1530	0.022	0.000	0.008(0.003)
p43	43	1.02e+05(4.67e+03)	5620	5637	5620	0.003	0.000	0.001(2.5e-4)
ry48p	48	2.62e+05(1.59e+04)	14422	14810	14446	0.027	0.002	0.012(0.003)
ft53	53	2.94e+05(1.58e+04)	6905	7236	6973	0.048	0.010	0.029(0.005)
ft70	70	4.73e+05(2.91e+04)	38673	39751	38744	0.028	0.002	0.017(0.003)

Table 6: Performance of MRAS on various ATSP problems based on 10 independent replications.

we performed 10 independent runs of the algorithm. In Table 6, N_{total} is the total number of tours generated (mean and standard error reported), H_{best} is the length of the true (optimal) shortest path, H_* and H^* are the worst and best solutions obtained out of 10 trials of the MRAS_1 algorithm, δ_* and δ^* are the respective relative errors for H_* and H^*, and δ is the relative error (mean and standard error reported). For all cases, $\varepsilon = 1$, the initial samples $N_0 = 1000$, $\rho_0 = 0.1$, $\lambda = 0.02$, $\alpha = 1.5$, $r = 0.1$, the stopping control parameters $d = 5$, $\tau = 0$, $N_{max} = 10N_c^2$, smoothing parameter $\upsilon = 0.5$, and the initial transition matrix $\widetilde{P}_0$ is initialized as a stochastic matrix whose $(i,j)th$ entry is proportional to the inverse of the (i,j)th entry of G, i.e., $\widetilde{P}_0(i,j) \propto \frac{1}{G_{i,j}}$ and $\sum_j \widetilde{P}_0(i,j) = 1\ \forall i$.

6. CONCLUSIONS

We have proposed a randomized optimization technique called Model Reference Adaptive Search (MRAS) for solving both continuous and discrete optimization problems. Highlights of the method include the following: (1) It is generic, requiring only a few mild regularity conditions on the underlying problem. (2) It is convergent w.p.1 to the set of ε-optimal solutions in a finite number of iterations and asymptotically to a global optimal solution. (3) It is insensitive to the choices of the initial solutions (parameter vectors), provided that the initial sampling variance is large enough. (4) It offers an alternative *general framework* for global optimization, based on which one can design and implement other efficient algorithms.

The MRAS algorithm demonstrated great promise on some preliminary examples, but practical implementation issues remain. For example, selection of the input parameters in our numerical experiments was based mainly on trial and error. For a given problem, how to determine a priori the most appropriate values of these parameters is an open issue. Designing an adaptive scheme to update these parameters during the search process may also enhance the convergence rate of the algorithm.

7. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[2] P. T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operation Research*, to appear, 2004.

[3] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. on Evolutionary Computation*, 1(1), 1997.

[4] F. Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.

[5] T. Homem-de Mello and R. Y. Rubinstein. Rare event estimation for static models via cross-entropy and importance sampling. *http://iew3.technion.ac.il/CE/pubs.php*, 2003.

[6] J. Hu, M. C. Fu, and S. I. Marcus. A model reference adaptive search algorithm for global optimization. *Operations Research*, Under review, 2005.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[8] D. P. Kroese, R. Y. Rubinstein, and S. Porotsky. The cross-entropy method for continuous multi-extremal optimization. *Operations Research*, Under review, 2004.

[9] H. Múhlenbein and 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, Springer Verlag, Berlin, 1996.

[10] R. Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 2:127–190, 1999.

[11] R. Y. Rubinstein. Combinatorial optimization, ants and rare events. In *In S. Uryasev and P. M. Pardalos, editors, Stochastic Optimization: Algorithms and Applications*, pages 304–358. Kluwer, 2001.

[12] R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. Springer Science+Business Media, Inc, New York, 2004.

[13] L. Shi and Ólafsson. Nested partitions method for global optimization. *Operations Research*, 48(3), 2000.

[14] M. Srinivas and L. M. Patnaik. Genetic algorithms: a survey. *IEEE Computer*, 27(6):17–26, 1994.

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