Model-Based Randomized Methods for Global Optimization

Michael C. Fu, Jiaqiao Hu, and Steven I. Marcus

***Abstract*— We survey some randomized search methods for global optimization that are based on sampling from an underlying probability distribution "model" on the solution space. In this approach, the probability model is updated iteratively after evaluating the performance of the samples at each iteration. Such model-based methods include estimation of distribution algorithms (EDAs), the cross-entropy method (CEM), and the recently proposed model reference adaptive search (MRAS).**

***Keywords*— Global optimization, Cross-entropy method, Estimation of distribution algorithm, Model reference adaptive search**

I. INTRODUCTION

Global optimization problems arise in a wide range of applications and are often extremely difficult to solve. In [33], solution methods are classified as being either *instance-based* or *model-based*. In *instance-based* methods, the search for new candidate solutions depends directly on previously generated solutions, e.g., simulated annealing (SA) [14], genetic algorithms (GAs) [8], [29], tabu search [7] and the recently proposed nested partitions (NP) method [27], [28].

On the other hand, in *model-based* algorithms, new candidate solutions are generated via an intermediate *probability model* that is iteratively updated. Thus, in general, each iteration of a model-based algorithm involves two separate phases:

1) Generate candidate solutions using the current probability model.
2) Update the probability model based on the generated candidate solutions, to direct the future search towards regions of "better" solutions.

In this paper, we focus on two model-based methods: the cross-entropy (CE) method [5], [21], [22], [23], [24] and model reference adaptive search (MRAS) [10], [11], [12]. We also discuss in broader terms estimation of distribution algorithms (EDAs) [17], [19], for which the CE method and MRAS could be viewed as specific approaches. Other related model-based approaches not reviewed here include annealing adaptive search [26],[31] and probability collectives [1], [30]; nor are swarm intelligence algorithms such as ant colony optimization [6] discussed.

This work was supported in part by the National Science Foundation under Grant DMI-0323220, and by the Air Force Office of Scientific Research under Grant FA95500410210.

M. Fu is with the Robert H. Smith School of Business and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA mfu@rhsmith.umd.edu

J. Hu is with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA jqhu@glue.umd.edu

S. Marcus is with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA marcus@eng.umd.edu

II. PROBLEM SETTING

Consider the following optimization problem:

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

where $\mathcal{X}$ is the solution space, H is a (univariate) real-valued function, and we assume (1) has a unique global optimal solution: i.e., there exists $x^* \in \mathcal{X}$ such that $H(x) > H(x^*)\ \forall x \neq x^*,\ x \in \mathcal{X}$.

An illustration of how a model-based method should work is shown in Fig. 1, where $H(x)$ is a simple one-variable function whose graph is shown in solid blue in the lower half of the figure. $H(x)$ has many local optima (e.g., near $x = -2, -5.7, +2.3$) and the global optimum at $x = 0$. The upper graph of Fig. 1 provides a possible snapshot of sequential distributions over the solution space (probability density functions) and samples from those distributions (indicated by the red $\times$ marks on the horizontal axis, but unable to be differentiated over time in the figure) that might be obtained from a single (short) run of a model-based algorithm. The solid (red) curve represents the initial distribution, which is relatively spread out over the solution space. Samples would be initially obtained from this distribution, and then used to update the distribution to the green dashed curve. This would be repeated to obtain the subsequent more peaked distributions shown in the graph.

III. THE PARADIGM SHIFT

In this section, we discuss the paradigm shift in going from instance-based approaches to the model-based framework. Clearly, both are ultimately interested in finding good candidate solutions. How these are generated is what differentiates approaches. The first conceptual leap from the oldest optimization approaches such as Newton's method (for local optimization) and later simulated annealing (for global optimization) is the use of a *population* of candidate solutions rather than iterating on a single solution. *Interaction* between solutions is crucial to these approaches; otherwise, one could simply run parallel versions (with multiple starting points) of the particular algorithm (e.g., simulated annealing). Thus, the framework of genetic algorithms, or more generally population-based evolutionary computation, consists of iterative algorithms with the following two phases:

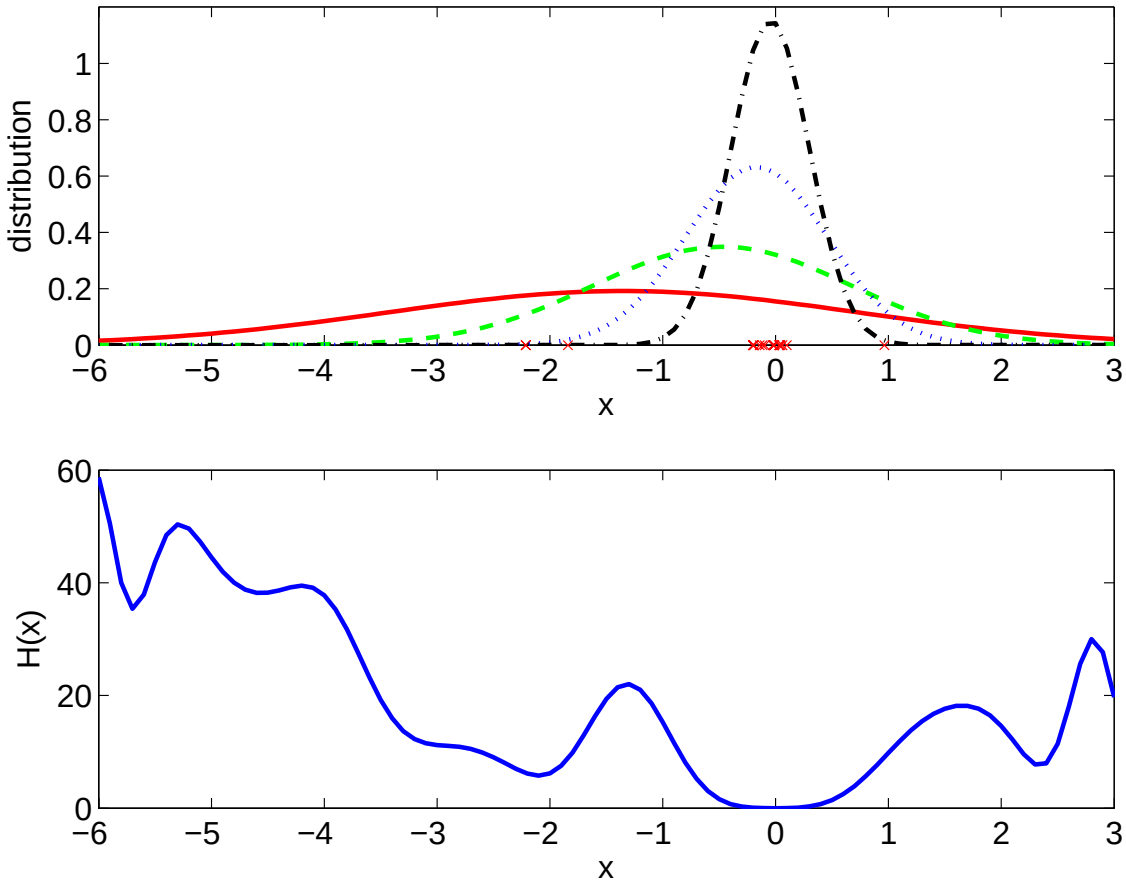


Fig. 1. Optimization via model-based methods

1) **Generation**:
   Given *population* $Y^{(k)}$,
   generate new candidate solutions
   via genetic *operators* on $Y^{(k)}$
   (e.g., crossover and mutation)
   $X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}$.
2) **Update**:
   population $Y^{(k+1)}$ based on
   selection among $Y^{(k)}$ and $X^{(k)}$;
   counter $k \leftarrow k + 1$.

Note that the two key steps are *generation* and *selection*.

In contrast, as introduced earlier, model-based approaches work with a *probability distribution* over the solution space, which can be viewed as a "model" of our best guess of the most promising regions. In this case, the iterative step consists instead of the following two phases:

1) **Generation**:
   Given *probability model* g_k,
   generate new candidate solutions (population)
   by *sampling* from g_k
   $X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}$.
2) **Update**:
   probability model $g_{(k+1)}$ based on $X^{(k)}$;
   counter $k \leftarrow k + 1$.

The big algorithmic question here is how the update in step 2 is carried out. Important practical implementation issues are the efficient construction or representation of the probability models g_k, and efficient sampling from g_k over the solution space $\mathcal{X}$. For example, if $\mathcal{X} = \Re^d$, then one possible probability distribution is the multivariate normal (Gaussian) distribution, which can be efficiently sampled from and represented relatively compactly by its mean vector and variance-covariance matrix. An even more frugal representation can be obtained by assuming independence among the individual dimensional components of a solution and thus working with univariate distributions.

The two previous approaches were presented as mutually exclusive, but a model-based framework that retains the instance-based population would look as follows:

1) **Generation**:
   Given probability model g_k and population $Y^{(k)}$,
   generate new candidate solutions by sampling
   from g_k and/or via genetic operators on $Y^{(k)}$
   $X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}$.
2) **Update**:
   population $Y^{(k+1)}$ and probability model $g_{(k+1)}$
   based on $Y^{(k)}$ and $X^{(k)}$;
   counter $k \leftarrow k + 1$.

IV. OVERVIEW OF APPROACHES

Estimation of distribution algorithms (EDAs) were introduced in the field of evolutionary computation in [19]. Like genetic algorithms (GAs), they work with a population of solutions; however, they differ in how candidate solutions are generated and subsequently propagated from generation to generation. As a model-based approach, generation of candidate solutions are carried out by sampling from the probability distribution model, eliminating the use of genetic operators such as crossover and mutation. The new population of candidate solutions is then used to update the probability model. The book [18] discusses EDAs in detail. The probability distributions in EDAs are often represented using graphical models, e.g., Bayesian networks.

The CE method was originally motivated by the problem of estimating rare event probabilities in simulation [21], before it was discovered that it could be adapted to solving combinatorial and continuous optimization problems [21], [22]. The book [24] summarizes many of the contribution The Web site `http://www.cemethod.org` contains the most up-to-date progress and references on the method. A very nice tutorial is given in [5].

The key ideas of CE are the following:

1) targeting an optimal (importance sampling) distribution concentrated only on the set of optimal solutions (i.e., zero variance),
2) working with a *parameterized* family of probability distributions;
3) optimizing the parameters iteratively by minimizing the *Kullback-Leibler (KL) divergence* between the parameterized distribution and the target optimal distribution.

The KL divergence between two probability density (or mass) functions f_1 and f_2 is defined as follows:

$$\mathcal{D}(f_1, f_2) := \int_{x\in\mathcal{X}} \ln \frac{f_1(x)}{f_2(x)} f_1(dx). \tag{2}$$

Since it is not symmetric (and also doesn't satisfy the triangle inequality), it is not a metric. If we let $g(\cdot)$ denote the (unknown) target distribution and $f(\cdot;\theta)$ the distribution parameterized by θ), then CE attempts to minimize

$$\mathcal{D}(g, f) = E_g \left[\ln \frac{g(X)}{f(X;\theta)} \right],$$

where X denotes a random variable having the distribution under which the expectation is indicated, i.e., in general, $E_g[H(X)] = \int_{\mathcal{X}} H(x) g(dx)$.

For the exponential family of distributions, the minimization can be carried out in analytical form, which makes the CE method very easy to implement efficiently.

Model reference adaptive search (MRAS) was introduced in [10]. It is similar to CE, in that it also works with parameterized distributions and updates the parameters by minimizing KL divergence. However, instead of having a fixed single target distribution, MRAS is a general framework that allows the user to specify the sequence of reference distributions. Thus, the key steps of MRAS are the following:

1) selecting a sequence of *reference* distributions $\{g_k\}$ with the desired convergence properties (e.g., the limit distribution being concentrated only on the set of optimal solutions),
2) working with a *parameterized* family of probability distributions $\{f(\cdot;\theta)\}$;
3) optimizing the parameters iteratively by minimizing the KL divergence between the parameterized distribution $f(\cdot;\theta)$ and the reference distribution $\{g_k\}$, i.e.,

$$\theta_k = \arg\min_{\theta} \mathcal{D}(g_k, f).$$

Again, for the exponential family of distributions, the minimization can be carried out in analytical form, which makes MRAS also easy to implement efficiently. The MRAS framework permits considerable flexibility in the choices of the reference models. By carefully analyzing and selecting the sequence of reference models, one can design different (perhaps even more efficient) versions. As we shall see, the sequence of reference distributions depends directly on the performance of the sampled candidate solutions. The instantiation of MRAS considered in this paper (to be presented shortly) uses a proportional selection scheme (which has also been used in EDAs for the probability models themselves, as opposed to the reference distributions here), and generates a sequence of reference distributions that guarantees improvement in the expected value sense. The papers [10], [11], [12] describe the MRAS method and its stochastic version (SMRAS), including detailed implementation of resulting algorithms and rigorous theoretical proofs of convergence (caution: the framework in those papers are in the **maximization** context), and provide some illustrative numerical studies on both continuous and combinatorial deterministic optimization problems (some of which will be presented later on in the numerical examples section), and stochastic optimization problems, which include some of the previous continuous deterministic functions with additive noise, as well as inventory control examples and (discrete) buffer allocation problems.

Remark. Order matters! In the MRAS framework, the sequence of implicit reference distributions $\{g_k\}$ is chosen first, because that allows one to analyze the convergence properties of a given instantiation based on the selection. Under fairly mild conditions, selecting a parameterized distribution from the exponential family and minimizing the KL divergence in updating the parameters will lead to the desired convergence, as long as the implicit reference distributions converge appropriately. Soon we will see that CE could be viewed as MRAS where steps (2) and (1) are interchanged, allowing g to depend on f.

A. Comparison between MRAS and CE

The basic algorithm for either CE or MRAS is shown in Fig. 2 (simply change "min" to "max" in step 2 for a

- **Initialize** $k \leftarrow 0, \theta_0, \hat{X}^*$.
- **Repeat until stopping criterion satisfied**
  1) **Generate candidate solutions** by sampling from $f(\cdot;\theta_k)$: $X_1^{(k)}, \ldots, X_N^{(k)}$.
  2) Compute $H(X_i^{(k)})\ \forall i$. Store the best thus far: $\hat{X}^* \leftarrow \arg\min_i\{H(\hat{X}^*), H(X_i^{(k)})\}$.
  3) Determine the "elite" candidate solutions among those newly generated: $\mathcal{X}_k^{elite} \subset \{X_i^{(k)}\}$.
  4) **Update parameters** θ_{k+1} of probability distribution based on $\mathcal{X}_k^{elite}$.
  5) $k \leftarrow k+1$.
- **Return** $\hat{X}^*$ as the estimated optimum.

Fig. 2. Basic CE and MRAS Algorithm Framework

maximization problem). It is clear that the CE method and MRAS share many similarities:

- Both work with a parameterized distribution, $f(x;\theta)$, most conveniently from the exponential family.
- Both minimize the Kullback-Leibler (cross-entropy) divergence between the parameterized distribution $f(x;\theta)$ and another unknown (implicit) distribution $g(x)$.
- Both iteratively update the parameter estimate θ_{k+1} based on some proportion of the samples taken in an iteration (determined from quantile estimates generally, or sometimes simply a fixed number in the CE method).

In sum, a primary advantage of the CE method and MRAS is that by using parameterized distributions, the difficult task of constructing the explicit probability models in EDAs is replaced by *implicit* use of reference models to guide the parameter updating procedure by way of minimizing the KL-divergence between the parameterized and reference models. The use of exponential families for the parameterized distributions further lends itself to computational tractability in executing the parameter update.

The primary difference between the two approaches has to do with the implicit distribution with which the KL-divergence is minimized is step 4 in Fig. 2. In CE, the target distribution is a *single fixed* distribution, which ideally is the optimal importance sampling measure for estimating $P(H(X) \leq \gamma)$ given by

$$g^*(x) = \frac{\mathbf{1}\{H(x) \leq \gamma\} f(x;\theta)}{E_f[\mathbf{1}\{H(X) \leq \gamma\}]}, \tag{3}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function.

On the other hand, MRAS works with a *sequence* of *reference* distributions, $\{g_k(x)\}$. The particular instantiation considered in [10] was the following:

$$g_k(x) = \frac{S(H(x)) g_{k-1}(x)}{\int_{\mathcal{X}} S(H(x)) g_{k-1}(dx)}, \tag{4}$$

where $S(\cdot)$ is a non-negative (to prevent negative probabilities) *decreasing* (increasing for maximization problems) function. This form of reference distributions can be viewed as an EDA with proportional selection scheme that weights the new density by the value of the objective function $H(x)$, i.e., more mass is being given to solutions with good performance. The resulting g_k have the property that the each iteration of (4) improves the expected performance, since

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

so solutions that have small values for H are given greater weight. Furthermore, one can also show that $\{g_k(\cdot),\ k = 0, 1, \ldots\}$ converges to a p.d.f. concentrated on the set of optimal solutions, so $\lim_{k\to\infty} E_{g_k}[H(X)] = H(x^*)$. The associated parameter update is the following:

$$\theta_{k+1} = \arg\max_{\theta\in\Theta} \frac{1}{N_k} \sum_{X \in \mathcal{X}_k^{elite}} \frac{[S(H(X))]^k}{f(X,\theta_k)} \ln f(X_i^{(k)}, \theta), \tag{5}$$

where $\{\bar{\gamma}_k\}$ is a (stochastic) sequence based on a quantile estimate, used to determine the elite samples used in the update:

$$\mathcal{X}_k^{elite} = \left\{ X_i^{(k)} : H(X_i^{(k)}) \leq \bar{\gamma}_{k+1} \right\}.$$

Returning to the CE method, in actual implementation, the parameter θ is also updated iteratively, so the optimal importance sampling measure in (3) is also changing, i.e., simply taking the sequence $\{g_k\}$ to be

$$g_k(x) = g^*(x;\theta_k),$$

CE can be cast in the MRAS framework as a particular instantiation, in which g_k *depends* on $f(\cdot;\theta_k)$, not a natural choice a priori in MRAS, where the reference distributions would generally be chosen separately and independently from the choice of parameterized distributions (see steps 1 and 2 in the description of MRAS). However, by taking this view of the CE method, the MRAS theory can be used to analyze the convergence properties of CE; see [10] for more details.

Another more subtle difference is the calculation of the elite samples by way of the quantile estimates. It turns out that the theoretical convergence of the CE method depends on this choice, and an inappropriate choice can lead to non-convergence of the algorithm; see [10] for more details.

$$\mu_{k+1} = \frac{\sum_{X \in \mathcal{X}_k^{elite}} X[S(H(X)]^k \exp(+0.5(X-\mu_k)^T \Sigma_k^{-1}(X-\mu_k))}{\sum_{X \in \mathcal{X}_k^{elite}} [S(H(X)]^k \exp(+0.5(X-\mu_k)^T \Sigma_k^{-1}(X-\mu_k))}, \tag{6}$$

$$\Sigma_{k+1} = \frac{\sum_{X \in \mathcal{X}_k^{elite}} (X-\mu_{k+1})(X-\mu_{k+1})^T [S(H(X)]^k \exp(+0.5(X-\mu_k)^T \Sigma_k^{-1}(X-\mu_k))}{\sum_{X \in \mathcal{X}_k^{elite}} [S(H(X)]^k \exp(+0.5(X-\mu_k)^T \Sigma_k^{-1}(X-\mu_k))}, \tag{7}$$

Fig. 3. MRAS Update Equations for Multivariate Normal Model

B. Example: Normal Distribution

For continuous optimization, the most commonly used parameterized distribution is the normal (Gaussian) distribution. For the standard CE method, independent univariate distributions are used, i.e., the parameters are vectors μ and σ representing the mean and standard deviation, and the parameter updates simply become the sample mean and sample variance over the elite candidates, i.e.,

$$\begin{aligned} \mu_{k+1} &= \bar{X}_k^{elite}, \\ \sigma_{k+1} &= s_k^{elite}, \end{aligned}$$

where $\bar{X}_k^{elite}$ and s_k^{elite} denote the sample mean and standard deviation over the elite samples of $\mathcal{X}_k^{elite} \subset \{X_1^{(k)}, \ldots, X_N^{(k)}\}$ in the kth iteration. The explicit update equations for the specific case where the MRAS parameterized model used is a multivariate Gaussian distribution,

$$f(x, \theta_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} e^{-(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)/2}, \tag{8}$$

parameterized by $\theta_k := (\mu_k;\ \Sigma_k)$, comprising its mean vector μ and variance-covariance matrix Σ, and the sequence of reference models follows (4), are provided in Fig. 3. A similar extension for the CE method would take the variance-covariance matrix to be the following:

$$\Sigma_{k+1} = \frac{1}{|\mathcal{X}_k^{elite}|} \sum_{X \in \mathcal{X}_k^{elite}} (X-\mu_{k+1})(X-\mu_{k+1})^T.$$

V. NUMERICAL EXAMPLES

We illustrate the performance of the standard CE method and the proportional reference model instantiation of MRAS based on (4) for some continuous and combinatorial optimization problems.

A. Implementation Issues

All examples are minimization problems, so we take $S(y) = \exp(-ry)$, where r is a problem-dependent tuning parameter. For maximization problems, the following adjustments would need to be made: $S(\cdot)$ *increasing* instead of decreasing; in steps 2, 3, and 4, lower tail quantile estimates changed to upper tail quantile estimates, along with "$\leq$" changed to "$\geq$" in step 3, (3), and (11).

Both the CE method and MRAS were implemented using a smoothed parameter updating procedure:

$$\widetilde{\theta}_{k+1} := \upsilon \theta_{k+1} + (1-\upsilon)\widetilde{\theta}_k, \quad 0 < \upsilon \leq 1,$$

where υ is the smoothing parameter, $f(x, \widetilde{\theta}_{k+1})$ (instead of $f(x, \theta_{k+1})$) is used in step 1 of the algorithm framework of Fig. 2 to generate new samples. For example, in the CE method, the update equations become the following:

$$\mu_{k+1} = \upsilon \bar{X}_k^{elite} + (1-\upsilon)\mu_k, \tag{9}$$

$$\sigma_{k+1} = \upsilon s_k^{elite} + (1-\upsilon)\sigma_k. \tag{10}$$

MRAS further mixes the current distribution with the initial distribution $f(x, \theta_0)$ using mixing parameter $0 < \lambda \leq 1$. MRAS also involves adaptively setting sample sizes according to the quantile estimates, with N_k defining the number of samples taken at iteration k. N_{min} is a lower limit on the number of samples required to updated the parameter. Fig. 4 gives the detailed MRAS algorithm used in the numerical examples. It should be clear that it is quite a bit more complicated than the CE method. For high-dimensional problems, the value of the p.d.f. in updating the parameter via (12) may be close to zero, so caution must be used to avoid numerical problems, e.g., in calculating the denominator terms in (6) and (7).

B. Continuous Optimization

For the continuous optimization problems, normal (Gaussian) distributions are used. The MRAS instantiation considered here uses the multivariate p.d.f. given by (8), whereas the CE method uses independent univariate p.d.f.s with mean vector μ_k and variance vector σ_k^2, using tuning parameters selected according to [15]. The probability model parameters updates for (12) in step 4 of Fig. 4 are carried out via (6) and (7) for MRAS and via (9) and (10) for the CE method. Both algorithms are compared with simulated annealing (SA), implemented using the tuning parameters suggested in [4].

Seven test functions were considered: H_1 and H_2 are low-dimensional functions with a few local optima separated by plateaus and relatively far apart; H_3 and H_4 are 20-dimensional badly-scaled functions; H_5 and H_6 are highly multimodal with the number of local optima increasing exponentially in the problem dimension; H_7 is

- **Initialize** $N_0 = 1000$, $\rho_0 = 0.1$, $N_{min} = 5d$ (d is dimension of $\mathcal{X}$).
  Specify $\varepsilon \geq 0$, $\alpha > 1$, $r > 0$, $0 < \lambda \leq 1$, $0 < \upsilon \leq 1$, 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 stopping rule satisfied**
  1. Generate i.i.d. $X_1^{(k)}, \ldots, X_{N_k}^{(k)} \sim \widetilde{f}(\cdot, \widetilde{\theta}_k) := (1-\lambda) f(\cdot, \widetilde{\theta}_k) + \lambda f(\cdot, \theta_0), \quad \widetilde{\theta}_k := (1-\upsilon)\theta_k + \upsilon \widetilde{\theta}_{k-1}$.
  2. Compute the lower sample ρ_k-quantile $\widetilde{\gamma}_{k+1}(\rho_k, N_k) := H_{(\lceil \rho_k N_k \rceil)}$,
     where $\lceil a \rceil$ = smallest integer greater than a, $H_{(i)} = i$th order statistic of $\{H(X_i^{(k)})\}$.
     Store the best thus far: $\hat{X}^* \leftarrow \arg\min_i \{H(\hat{X}^*), H(X_i^{(k)})\}$.
  3. **If** $k = 0$ **or** $\widetilde{\gamma}_{k+1}(\rho_k, N_k) \leq \bar{\gamma}_k + \frac{\varepsilon}{2}$,
     then 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 $\widetilde{\gamma}_{k+1}(\bar{\rho}, N_k) \leq \bar{\gamma}_k + \frac{\varepsilon}{2}$.
     If such a $\bar{\rho}$ exists, **then** set $\bar{\gamma}_{k+1} \leftarrow \widetilde{\gamma}_{k+1}(\bar{\rho}, N_k), \; \rho_{k+1} \leftarrow \bar{\rho}, \; N_{k+1} \leftarrow N_k$.
     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

$$\mathcal{X}_k^{elite} = \left\{ X_i^{(k)} : H(X_i^{(k)}) \leq \bar{\gamma}_{k+1} \right\}. \tag{11}$$

  4. If $|\mathcal{X}_k^{elite}| > N_{min}$, then update parameter:

$$\theta_{k+1} = \arg\max_{\theta \in \Theta} \frac{1}{N_k} \sum_{X \in \mathcal{X}_k^{elite}} \frac{\exp(-rkH(X))}{\widetilde{f}(X, \widetilde{\theta}_k)} \ln f(X, \theta). \tag{12}$$

  5. Set $k \leftarrow k + 1$.
- **Return** $\hat{X}^*$ as the estimated optimum.

Fig. 4. MRAS Algorithm used in numerical examples (for minimization problems)

both badly scaled and highly multimodal. To get an idea of some of these characteristics, Fig. 5 displays four of these functions in two dimensions. For shorthand, we denote $H^* = H(x^*)$ as the objective function minimal value.

1) Dejong's 5^{th} function x^*=$(-32, -32)$, $H^* \approx .998$:

$$H_1(x) = \Big[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{j,i})^6} \Big]^{-1},$$

where $a_{j,1} = \{-32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32\}$, $a_{j,2} = \{-32, -32, -32, -32, -32, -16, -16, -16, -16, -16, 0, 0, 0, 0, 0, 16, 16, 16, 16, 16, 32, 32, 32, 32, 32\}$; 24 local minima.

2) Shekel's function $x^* \approx (4, 4, 4, 4)$, $H^* \approx -10.153$:

$$H_2(x) = \sum_{i=1}^{5} \Big((x - a_i)^T (x - a_i) + c_i \Big)^{-1},$$

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 $c = (0.1, 0.2, 0.2, 0.4, 0.4)$.

(3) Rosenbrock function (d=20) x^*=$(1, \ldots, 1)$, H^*=0:

$$H_3(x) = \sum_{i=1}^{d-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2.$$

(4) Powel singular function (d=20) x^*=$(0, ..., 0)$, H^*=0:

$$H_4(x) = \sum_{i=2}^{d-2} \big[(x_{i-1} + 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4 \big].$$

(5) Trigonometric (d=20) x^*=$(0.9, ..., 0.9)$, H^*=1:

$$H_5(x) = 1 + \sum_{i=1}^{d} 8 \sin^2 \big(7(x_i - 0.9)^2 \big) + 6 \sin^2 \big(14(x_i - 0.9)^2 \big) + (x_i - 0.9)^2.$$

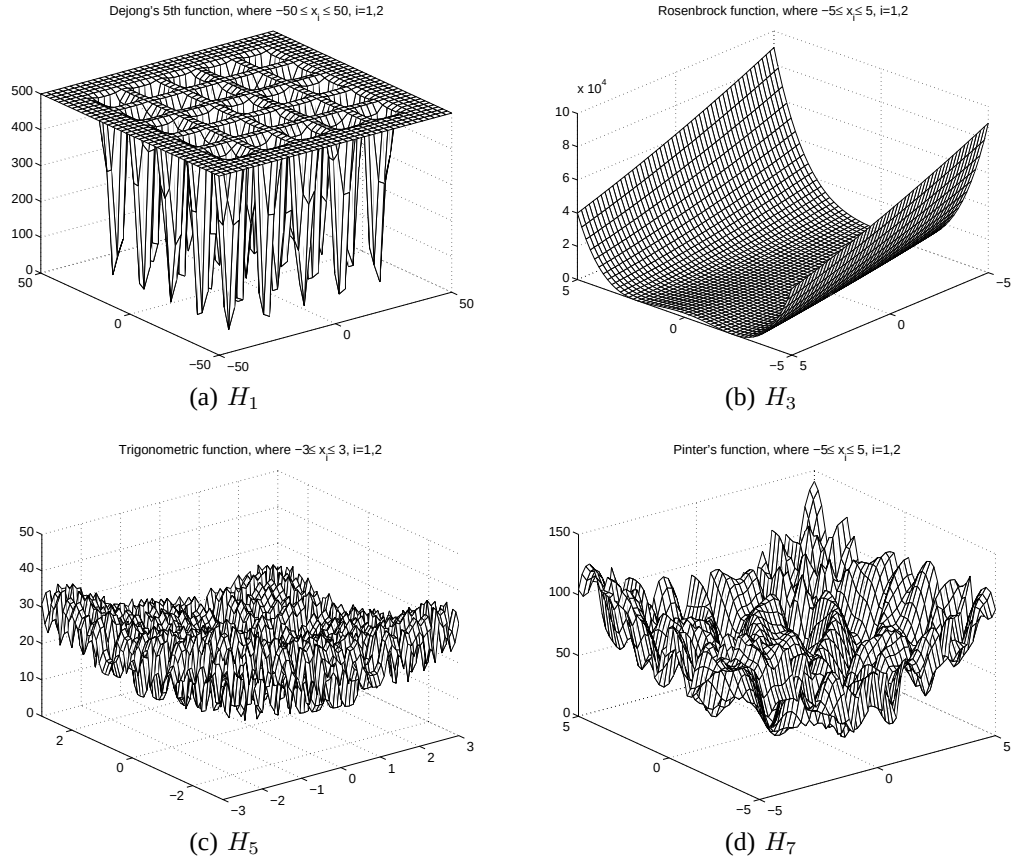
(6) Griewank function (d=20) $x^* = (0, \ldots, 0)$, $H^* = 0$:

$$H_6(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\Big(\frac{x_i}{\sqrt{i}} \Big) + 1.$$

(7) Pintér's function (d=20) $x^* = (0, \ldots, 0)$, $H^* = 0$:

$$H_7(x) = \sum_{i=1}^{d} i x_i^2 + \sum_{i=1}^{d} 20i \sin^2 \big(x_{i-1} \sin x_i - x_i + \sin x_{i+1} \big) + \sum_{i=1}^{d} i \log_{10}\big(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2 \big),$$

where $x_0 = x_d$, $x_{d+1} = x_1$ (cf. [20]).



(a) H_1 (b) H_3

(c) H_5 (d) H_7

Fig. 5. Selected test problems in two dimensions, (a) H_1: Dejong's 5th; (b) H_3: Rosenbrock; (c) H_5: Trigonometric; (d) H_7: Pintér.

TABLE I

PERFORMANCE COMPARISON FOR CONTINUOUS TEST FUNCTIONS

Results based on 50 independent replication runs (standard errors in parentheses),
where $\bar{H}_i^*$ = mean of $H_i(\cdot)$ at best solution visited by algorithm, M_ε = # replications ε-optimal solution found;
components of initial solution (components of mean vector μ_0 for CE and MRAS) randomly selected from $U(-50, 50)$,
all variances for CE and MRAS initially set to 500 (covariances 0);
MRAS: $\varepsilon = 10^{-5}, \lambda = 0.01, \alpha = 1.1, r = 10^{-4}, \upsilon = 0.2$; CE: $N = 2000, \rho = 0.01$;
SA: initial temperature $T = 50000$, temperature reduction factor $r_T = 0.85$, neighborhood $\mathcal{N}(x) = \{y : \max_{1 \le i \le d} |x_i - y_i| \le 1\}$.

		MRAS		CE ($\upsilon = 0.7$)		CE ($\upsilon = 0.2$)		SA	
	$H_i(x^*)$	$\bar{H}_i^*$	M_ε	$\bar{H}_i^*$	M_ε	$\bar{H}_i^*$	M_ε	$\bar{H}_i^*$ (stderr)	M_ε
H_1	0.998	0.998 (3.0e-07)	50	2.26 (0.36)	31	0.998 (1.3e-08)	50	7.43 (0.96)	8
H_2	-10.153	-10.15 (3.18e-07)	50	-8.02 (0.45)	34	-9.94 (0.13)	0	-7.25 (0.39)	2
H_3	0	11.77 (0.54)	0	27.87 (3.43)	0	15.90 (1.7e-02)	0	203.7 (11.34)	0
H_4	0	2.8e-10 (2.1e-11)	50	1.0e+04 (4.0e+03)	3	2.9e-06 (1.5e-07)	50	65.90 (2.96)	0
H_5	1	1.59 (0.13)	24	1.00 (00e-00)	50	1.00 (5.7e-12)	50	65.17 (1.22)	0
H_6	0	4.0e-03 (7.0e-04)	28	1.5e-04 (1.5e-04)	49	2.2e-12 (3.9e-13)	50	0.15 (0.04)	0
H_7	0	3.5e-09 (6.5e-10)	50	2.26 (9.7e-02)	0	6.2e-04 (3.3e-05)	0	1.7e+03 (50.8)	0

Table I summarizes the results for the seven cases, based on 50 independent replication runs for each algorithm and a fixed amount of computational effort, where the total number of function evaluations (i.e., sample size) is set to 50,000 for the first two cases and 400,000 for the other five 20-dimensional cases. For H_3, none of these three algorithms found ε-optimal solutions, with SA performing the worst. For the highly multimodal functions H_5 and H_6, CE ($\upsilon = 0.7$) works best. The behavior of MRAS can be explained by looking at the parameter updating equations (6) and (7). Since the values of H_5 and H_6 at local minima near the optimum are very close to each other, the parameter updating in MRAS is dominated by the density function in the denominator, especially when the iteration counter k is small. H_7 contains both a badly-scaled quadratic term and some badly-scaled noise terms. For this function, SA does not seem to be competitive at all.

The experimental results indicate that the proportional reference model instantiation of MRAS is better adapted to optimization of badly scaled multimodal problems, whereas the CE method works best on problems that are well-scaled and contain a large number of local optima. Both outperform simulated annealing in the long run, though the local search portion of simulated annealing sometimes allow it to find better solutions (locally optimal) early on [10].

C. Combinatorial Optimization

In this section, we describe the approach used to solve *asymmetric* (where the distance from city i to city j is not necessarily the same as the distance from city j to city i) traveling salesman problems (ATSPs) using MRAS or the CE method, following [23], [5]. We then illustrate the typical performance of MRAS on various ATSPs; the CE method performs comparably, so no explicit comparisons are presented.

Each ATSP is specified by an N_c by N_c distance matrix, where the (i,j)th entry gives the distance from city i to city j, and N_c is the number of cities. A candidate solution is an *admissible* tour, a path that visits all N_c of the cities and returns to the starting city. The goal is to find the shortest path, so the objective function H is the length of an admissible tour. In the model-based setting, the parameter "vector" to be optimized is a transition matrix $\theta(i,j)$ whose (i,j)th element specifies the *probability* of transitioning from city i to city j. Thus, an iteration of the algorithm executes the following two steps:

(i) Generate admissible tours sequentially via θ — from city i, the next city j is chosen with (renormalized) probability $\theta(i,j)$, for all j *not yet visited on the tour* — and calculate path lengths.

(ii) Update transition matrix based on the sample tours.

"Convergence" would be constituted by $\theta(i,j)$ going to 1 for a set of city-pairs that form an admissible tour.

To update the transition matrix at each iteration, the model p.m.f. $f(\cdot,\theta_k)$ on the solution space is parameterized by the transition matrix θ_k and is given by

$$f(x,\theta_k) = \prod_{l=1}^{N_c} \sum_{i,j}^{N_c} \theta_k(i,j) \mathbf{1}\left\{x \in \mathcal{X}_{i,j}(l)\right\},$$

where $\mathcal{X}_{i,j}(l)$ is the set of all tours in $\mathcal{X}$ such that the lth transition is from city i to city j. After generating i.i.d. (admissible) sample tours $X_1^{(k)}, \ldots, X_{N_k}^{(k)}$ from $\widetilde{f}(\cdot,\widetilde{\theta}_k)$ in step 1 of the algorithm, steps 2 and 3 are carried out to find the elite samples $\mathcal{X}_{i,j}$ via (11), and the new transition matrix is updated in step 4 via (12) as

$$\theta_{k+1}(i,j) = \frac{\displaystyle\sum_{X \in \mathcal{X}_k^{elite}} \frac{e^{-rkH(X)}}{\widetilde{f}(X,\widetilde{\theta}_k)} \mathbf{1}\left\{X \in \mathcal{X}_{i,j}\right\}}{\displaystyle\sum_{X \in \mathcal{X}_k^{elite}} \frac{e^{-rkH(X)}}{\widetilde{f}(X,\widetilde{\theta}_k)}},$$

where $\mathcal{X}_{i,j}$ represents the set of tours in which the transition from city i to city j is made. CE would use the simpler $\theta_{k+1}(i,j) = \left|\{X \in \mathcal{X}_{i,j} : X \in \mathcal{X}_k^{elite}\}\right| / \left|\mathcal{X}_k^{elite}\right|$.

The performance of the MRAS algorithm is reported in Table II. For each problem instance, 10 independent replications of the algorithm were run. The algorithm terminates once $N_k > 10N_c^2$ or the following condition is satisfied: $\max_{1 \le i \le 5} |\bar{\gamma}_k - \bar{\gamma}_{k-i}| = 0$. The algorithm finds very good solutions relatively quickly; however, general global optimization approaches such as MRAS and CE would require considerable tuning to be truly competitive with algorithms specially designed for TSPs, since straightforward implementations of MRAS and CE do not exploit problem structure.

VI. CONCLUDING REMARKS

Model-based methods such as EDAs, the CE method, and MRAS are very promising randomized methods for global optimization, both in discrete and continuous solution spaces. Some directions worth pursuing include the following:

- Connections. The CE method was originally motivated by an estimation problem, so "filtering" estimation literature from control theory should be quite relevant. In particular the approach of particle filtering also involves the two steps of sampling from a distribution and then updating the probability distribution via Bayes rule. Connections between model-based methods and particle filtering would thus seem to be a fruitful line of research to investigate.
- Applications. A particular application of interest is cluster analysis, where it is argued in [13] that global optimization approaches have the potential to provide substantial performance improvements over traditional statistical approaches such as the expectation-maximization (EM) algorithm. In [13], a GA was applied. Initial work in this area using CE and MRAS includes [2], [16], [9]. Another application area of more general interest is the use of these methods in solving sequential decision making under uncertainty, specifically Markov decision processes; see [3] and associated papers in preparation.
- Theory. EDAs have relatively little mathematical theory behind them in terms of rigorous convergence proofs (cf. [32] for a convergence result under the infinite population assumption), because the focus has been on computational results and algorithms for specific applications or small examples. The CE method does have some convergence results (e.g., [22]), but again the focus has been on computation, where numerical experiments have shown it to be a particularly effective method for many problems. On the other hand, MRAS is firmly grounded in mathematically rigorous theory, and the framework provides a natural setting in which to explore algorithms from a theoretical perspective by considering different sequences of reference distributions.

TABLE II

PERFORMANCE FOR ATSPS

(taken from `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95`)

Results based on 10 independent replication runs (standard errors in parentheses);

N_{total} = # tours generated, H_{best} = length of shortest path, H_* and H^* worst and best solutions obtained, δ_* and δ^* respective relative errors, δ = mean relative error;

$\varepsilon = 1, \lambda = 0.02, \alpha = 1.5, r = 0.1, \upsilon = 0.5$, $\theta_0(i,j) \propto$ inverse of the (i,j)th entry of the distance matrix.

ATSP	N_c	N_{total}	H_{best}	H_*	H^*	δ_*	δ^*	δ
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)

REFERENCES

[1] S. Bieniawski, D.H. Wolpert, and I. Kroo 2004. Discrete, Continuous, and Constrained Optimization Using Collectives AIAA Paper 2004-4580. Presented at the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.

[2] Z. Botev and D.P. Kroese 2004. Global Likelihood Optimization via the Cross-Entropy Method with an Application to Mixture Models. *Proceedings of the 2004 Winter Simulation Conference*, 529-535.

[3] H.S. Chang, M.C. Fu, J. Hu, and S.I. Marcus 2006. *Simulation-Based Methods for Markov Decision Processes*. Springer, forthcoming.

[4] A. Corana, M. Marchesi, C. Martini, and S. Ridella 1987. Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm. *ACM Trans. Mathematical Software*, Vol.13, 262-280.

[5] P. T. De Boer, D. P. Kroese, S. Mannor, and R.Y. Rubinstein 2005. A Tutorial on the Cross-Entropy Method. *Annals of Operation Research*, Vol.134, 19-67.

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

[7] F.W. Glover 1990. Tabu Search: A Tutorial. *Interfaces*, Vol.20, 74-94.

[8] D.E. Goldberg 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley.

[9] J.W. Heath, M.C. Fu, and W. Jank 2006. Global Solutions to Model-Based Clustering. manuscript.

[10] J. Hu, M.C. Fu, and S.I. Marcus 2005. A Model Reference Adaptive Search Method for Global Optimization. *Operations Research*, forthcoming.

[11] J. Hu, M.C. Fu, and S.I. Marcus 2005. A Model Reference Adaptive Search Method for Stochastic Global Optimization. *Mathematics of Operations Research*, submitted.

[12] J. Hu, M.C. Fu, and S.I. Marcus 2005. Simulation Optimization using Model Reference Adaptive Search. *Proceedings of the 2005 Winter Simulation Conference*, 811-18.

[13] W. Jank 2006. The EM Algorithm, Its Randomized Implementation and Global Optimization: Some Challenges and Opportunities for Operations Research. *Topics in Modeling, Optimization, and Decision Technologies: Honoring Saul Gass' Contributions to Operations Research* (tentative title). F.B. Alt, M.C. Fu and B.L. Golden, editors, Kluwer. forthcoming.

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

[15] D. P. Kroese, R.Y. Rubinstein and S. Porotsky 2004. The Cross-Entropy Method for Continuous Multi-extremal Optimization. *Operations Research*. Under review.

[16] D. Kroese, R.Y. Rubinstein and T. Taimre 2004. Application of the Cross-Entropy Method to Clustering and Vector Quantization. manuscript.

[17] P. Larrañaga, R. Etxeberria, J.A. Lozano, B. Sierra, I. Iñza, and J.M. Peña 1999. A Review of the Cooperation Between Evolutionary Computation and Probabilistic Graphical Models. *Proceedings of the Second Symposium on Artificial Intelligence. Adaptive Systems*. CIMAF 99. Special Session on Distributions and Evolutionary Computation, 314-324.

[18] P. Larrañaga and J.A. Lozano 2001. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Springer.

[19] H. Mühlenbein and G. Paaß. 1996. 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*, Springer, 178-187.

[20] J.D. Pintér 1996. *Global Optimization in Action.* Kluwer.

[21] R.Y. Rubinstein 1997. Optimization of Computer Simulation Models with Rare Events. *European Journal of Operations Research*, Vol.99, 89-112.

[22] R.Y. Rubinstein 1999. The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology and Computing in Applied Probability*, Vol.2, 127-190.

[23] R.Y. Rubinstein 2001. Combinatorial Optimization, Ants and Rare Events. In S. Uryasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, 304-358, Kluwer.

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

[25] R.Y. Rubinstein and A. Shapiro 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.* John Wiley & Sons.

[26] Y. Shen 2005. *Annealing Adaptive Search with Hit-and-Run Sampling Methods for Global Optimization.* Ph.D. Thesis, Department of Industrial Engineering, University of Washington, Seattle.

[27] L. Shi and Ólafsson, S. 2000a. Nested Partitions Method for Global Optimization. *Operations Research*, Vol.48, 390-407.

[28] L. Shi and Ólafsson, S. 2000b, Nested Partitions Method for Stochastic Optimization. *Methodology and Computing in Applied Probability*, Vol.2, 271-291.

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

[30] D.H. Wolpert 2004. Finding Bounded Rational Equilibria Part I: Iterative Focusing, *Proceedings of the Eleventh International Symposium on Dynamic Games and Applications.*

[31] Z.B. Zabinsky 2003. *Stochastic Adaptive Search for Global Optimization.* Kluwer.

[32] Q. Zhang, H. Mühlenbein 2004. On the Convergence of a Class of Estimation of Distribution Algorithm. *IEEE Trans. on Evolutionary Computation*, Vol.8, 127-136.

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