Evolutionary Optimization by Distribution Estimation with Mixtures of Factor Analyzers

Dong-Yeon Cho, Byoung-Tak Zhang
Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
{dycho, btzhang}@bi.snu.ac.kr

Abstract - Evolutionary optimization algorithms based on the probability models have been studied to capture the relationship between variables in the given problems and finally to find the optimal solutions more efficiently. However, premature convergence to local optima still happens in these algorithms. Many researchers have used the multiple populations to prevent this ill behavior since the key point is to ensure the diversity of the population. In this paper, we propose a new estimation of distribution algorithm by using the mixture of factor analyzers (MFA) which can cluster similar individuals in a group and explain the high order interactions with the latent variables for each group concurrently. We also adopt a stochastic selection method based on the evolutionary Markov chain Monte Carlo (eMCMC). Our experimental results support that the presented estimation of distribution algorithms with MFA and eMCMC-like selection scheme can achieve better performance for continuous optimization problems.

I. Introduction

Evolutionary optimization techniques called estimation of distribution algorithms (EDAs) use the probability distribution models constructed from the good solutions of the population to generate new offsprings. That is, these algorithms replace crossover and mutation operators by sampling candidate solutions from the probability distribution. Thus, the critical part of EDAs is to estimate this distribution so accurately that it can capture the building-block structure of the given problem and guide further searches toward the optimal point.

For the purpose of probability model building, many researchers have applied various methods from the simplest ones where each variable in a problem is assumed to be independent, through the ones that consider only some pairwise interaction, to the complex ones which can represent the high-order dependencies. More detail reviews of existing EDAs are presented in [1] and [2].

However, premature convergence to local optima still happens in EDAs for many optimization problems since they only consider better individuals in the current population to build the probability model, which is not enough to estimate exact distributions. Recently, some sophisticated EDAs have been proposed to deal with this problem. The main idea of these EDAs is to use multiple populations or to create sub-populations for maintaining diverse individuals. Pelikan and Goldberg [3] empirically showed that clustering of the parent population by k-means algorithm and processing each cluster separately with Bayesian networks alleviate the problem of symmetry for the binary representations.

Another way to handle several populations in EDAs is to use mixture models. The usefulness of mixture models in the data analysis is validated by a number of papers about mixture applications [4]. A mixture model is able to model very complex distributions through a proper choice of its components to represent accurately the local areas of the support of the true distribution. Particularly for the evolutionary algorithms based on probabilistic model in continuous case, the normal mixtures of density functions have been successfully applied in [5], [6], and [7]. For the binary problems, Santana et al. [8] introduced a factorized distribution algorithm based on a mixture of trees distribution.

While most EDAs tried to explicitly represent the relationship between variables in the problems, we proposed a new type of EDA in our previous work [9] where latent variable models such as Helmholtz machine and probabilistic principal component analysis were used for capturing the relationship. Latent variable model provides a powerful approach to probabilistic model building by supplementing a set of directly observed variables with additional latent, or hidden, variables [10]. By defining a joint distribution over visible and latent variables, the corresponding distribution of the observed variables is then obtained by marginalization. This allows relatively complex distributions to be expressed in terms of more tractable joint distributions over the expanded variable

0-7803-7282-4/02/$10.00 ©2002 IEEE

space. In addition, it is so easy to sample new data from the estimated distribution since latent variable models are generative ones. More comprehensive explanation and experimental results for our continuous EDAs can be found in [11] and [12].

In this paper, we improve our previous work into a mixture version of continuous EDAs with latent variable models. For this purpose, we make use of the mixture of factor analyzers (MFA) proposed by Ghahramani and Hinton [13]. MFA is a statistical method which concurrently performs clustering and, within each cluster, modeling the covariance structure of high dimensional data using a small number of latent variables. We also adopt a new selection method which was used in the evolutionary Markov chain Monte Carlo (eMCMC) [14]. Here, the candidate individuals are selected according to a probability instead of replacing deterministically the old population with new one.

This paper is organized as follows. In section II, we explain the basic concept of MFAs. Section III presents the EDAs with MFAs for the continuous domain. Section IV reports the experimental results and analysis for some benchmark functions. Finally, conclusions of this study are drawn in section V.

II. Theoretical Framework of MFAs

A. Latent Variable Models

Latent variable models [10] try to represent the distribution $p(\mathbf{x})$ of d dimensional real-valued data vector $\mathbf{x}$ in terms of a q dimensional latent variables $\mathbf{z}$, where q is usually smaller than d. The first step for this task is to decompose the joint distribution $p(\mathbf{x}, \mathbf{z})$ into the product of the marginal distribution $p(\mathbf{z})$ of the latent variables and the conditional distribution $p(\mathbf{x}|\mathbf{z})$ of the data given the latent variables. The key assumption of this model is that the observed variables x_i are conditionally independent given the values of latent variables $\mathbf{z}$, so that the joint distribution is

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = p(\mathbf{z}) \prod_{i}^{d} p(x_i|\mathbf{z}).$$

Next, the conditional distribution $p(\mathbf{x}|\mathbf{z})$ is expressed by a mapping from latent variables to data variables with a noise as follow

$$\mathbf{x} = \mathbf{y}(\mathbf{z}; \mathbf{\Lambda}) + \boldsymbol{\epsilon},$$

where $\mathbf{y}(\mathbf{z}; \mathbf{\Lambda})$ is a function of the latent variable $\mathbf{z}$ with parameter $\mathbf{\Lambda}$, and $\boldsymbol{\epsilon}$ is a $\mathbf{z}$-independent noise process.

Then, we can have the final model for the distribution $p(\mathbf{x})$ of the data by marginalizing over the latent variables as follows

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}.$$

Generally, this integration may be impossible except for specific forms of the distributions $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$.

B. Factor Analysis

Standard factor analysis [15] is one of the simplest latent variable models and based on a linear mapping $\mathbf{y}(\mathbf{z}; \mathbf{\Lambda})$ so that $\mathbf{x} = \mathbf{\Lambda}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, where the latent variables $\mathbf{z}$ have a zero mean, unit covariance Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{\mu}$ is the mean vector of the whole data, i.e., $\boldsymbol{\mu} = \sum_{i=1}^{N} \mathbf{x}_i/N$, and the noise model is a zero mean Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{\Psi})$ with diagonal covariance matrix $\mathbf{\Psi}$. Here the diagonality of $\mathbf{\Psi}$ implies that the observed variables are independent given the latent variables. From this formulation, we can easily show that the data $\mathbf{x}$ has also a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Psi} + \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}})$. Thus the final goal of factor analysis is to find the factor loading matrix $\mathbf{\Lambda}$ and error covariance matrix $\mathbf{\Psi}$ which best describe the structure of data $\mathbf{x}$.

Although there is no closed form solution for $\mathbf{\Lambda}$ and $\mathbf{\Psi}$, we can obtain the maximum likelihood estimation for these parameters by using the Expectation-Maximization (EM) algorithm [16]. The EM algorithm is an iterative technique which is broadly applicable to maximum likelihood estimation for models in which there is missing information. In FAs, the incomplete data are the values of the latent variables $\mathbf{z}$. Rubin and Thayer [17] formulated the EM algorithm for factor analysis as followings.

First, in the E-step, we compute expectation of the complete data log-likelihood given the observed data $\mathbf{x}_i$ and the current estimated values of the parameters. Hence, we can find the conditional expected values and the second moments of latent variables over the distribution $p(\mathbf{z}_i|\mathbf{x}_i, \mathbf{\Lambda}, \mathbf{\Psi})$ as follows:

$$\begin{aligned} E\{\mathbf{z}_i|\mathbf{x}_i\} &= \boldsymbol{\beta}(\mathbf{x}_i - \boldsymbol{\mu}), \\ E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}|\mathbf{x}_i\} &= \mathrm{Var}\{\mathbf{z}_i|\mathbf{x}_i\} + E\{\mathbf{z}_i|\mathbf{x}_i\}E\{\mathbf{z}_i|\mathbf{x}_i\}^{\mathrm{T}} \\ &= \mathbf{I} - \boldsymbol{\beta}\mathbf{\Lambda} + \boldsymbol{\beta}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\beta}^{\mathrm{T}}, \end{aligned}$$

where $\boldsymbol{\beta} = \mathbf{\Lambda}^{\mathrm{T}}(\mathbf{\Psi} + \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}})^{-1}$. These statistics follow from the posterior distribution of $\mathbf{z}_i$ which can be easily proved to be Gaussian distribution

$$\mathcal{N}(\mathbf{\Lambda}^{\mathrm{T}}(\mathbf{\Psi} + \mathbf{\Lambda}\mathbf{\Lambda}^{\mathrm{T}})^{-1}(\mathbf{x}_i - \boldsymbol{\mu}), (\mathbf{\Lambda}^{\mathrm{T}}\mathbf{\Psi}^{-1}\mathbf{\Lambda} + \mathbf{I})^{-1}),$$

and the well-known matrix inversion lemma.

In the M-step, the expectation of log-likelihood is maximized with respect to parameters $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ by differentiating it and setting the derivatives to zero. This gives

the new parameter estimates

$$\mathbf{\Lambda}^{\text{new}} = \left[\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})E\{\mathbf{z}_i|\mathbf{x}_i\}^{\mathrm{T}}\right]\left[\sum_{i=1}^{N}E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}|\mathbf{x}_i\}\right]^{-1},$$

$$\mathbf{\Psi}^{\text{new}} = \frac{1}{N}diag\left[\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}} - \mathbf{\Lambda}^{\text{new}}E\{\mathbf{z}_i|\mathbf{x}_i\}(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}\right],$$

where the *diag* operator sets all elements of a matrix except the diagonal elements to zero. These two steps are repeated sequentially until the algorithm converges.

C. Mixture of Factor Analyzers

In a mixture of M factor analyzers indexed by $j = 1, \ldots, M$, we consider the following mixture distribution

$$p(\mathbf{x}) = \sum_{j=1}^{M}\pi_j p(\mathbf{x}|j) = \sum_{j=1}^{M}\int p(\mathbf{x}|\mathbf{z}, j)p(\mathbf{z}|j)p(j)d\mathbf{z}, \quad (1)$$

where $p(\mathbf{x}|\mathbf{z}, j)$ is a single factor analyzer model, therefore it can represent the covariance structure in a different part of data space with a Gaussian distribution $\mathcal{N}(\mathbf{\Lambda}_j\mathbf{z} + \boldsymbol{\mu}_j, \mathbf{\Psi})$ and $\pi_j = p(j)$ is the corresponding mixing proportion which satisfies the constraint $\sum_{j=1}^{M}\pi_j = 1$. As in a single factor analysis, the factors are all assumed to be distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$, thus $p(\mathbf{z}|j) = p(\mathbf{z})$.

We can estimate the parameters $\{(\boldsymbol{\mu}_j, \mathbf{\Lambda}_j, \pi_j)_{j=1}^{M}, \mathbf{\Psi}\}$ of a MFA with the EM algorithm similarly to FA. In this case, however, there is another missing information as well as the values of latent variables. That is, we cannot know which model is responsible for generating each data point $\mathbf{x}_i$. By using indicator variable d_{ij} that denotes whether $\mathbf{x}_i$ originates from the component density j, the complete data log-likelihood in the EM algorithm for MFA can be formulated as the following form

$$L_c = \sum_{i=1}^{N}\sum_{j=1}^{M}d_{ij}\ln\{\pi_j p(\mathbf{x}_i, \mathbf{z}_{ij})\}.$$

Now, in the E-step, we compute the expectation of L_c with observed data and previous values of the parameters. Then the probability h_{ij} with which the component j generates data point i is calculated as

$$h_{ij} = E\{d_{ij}|\mathbf{x}_i\} = \frac{\pi_j p(\mathbf{x}_i|j)}{\sum_{l=1}^{M}\pi_l p(\mathbf{x}_i|l)} = \frac{\pi_j|\mathbf{C}_j|^{-1/2}\exp\{-(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}\mathbf{C}_j^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_j)/2\}}{\sum_{l=1}^{M}\pi_l|\mathbf{C}_l|^{-1/2}\exp\{-(\mathbf{x}_i - \boldsymbol{\mu}_l)^{\mathrm{T}}\mathbf{C}_l^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_l)/2\}}, \quad (2)$$

where $\mathbf{C}_j = \mathbf{\Psi} + \mathbf{\Lambda}_j\mathbf{\Lambda}_j^{\mathrm{T}}$. We can also obtain the expected value of latent variables

$$E\{\mathbf{z}_{ij}|\mathbf{x}_i\} = \boldsymbol{\beta}_j(\mathbf{x}_i - \boldsymbol{\mu}_j),$$
$$E\{\mathbf{z}_{ij}\mathbf{z}_{ij}^{\mathrm{T}}|\mathbf{x}_i\} = \mathbf{I} - \boldsymbol{\beta}_j\mathbf{\Lambda}_j + \boldsymbol{\beta}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\beta}_j^{\mathrm{T}}, \quad (3)$$

where $\boldsymbol{\beta}_j = \mathbf{\Lambda}_j^{\mathrm{T}}(\mathbf{\Psi} + \mathbf{\Lambda}_j\mathbf{\Lambda}_j^{\mathrm{T}})^{-1}$.

Next, in the M-step, we re-estimate the values of parameters as in the following equations

$$\pi_j^{\text{new}} = \frac{1}{N}\sum_{i=1}^{N}h_{ij},$$

$$\tilde{\mathbf{\Lambda}}_j^{\text{new}} = \left[\sum_{i=1}^{N}h_{ij}\mathbf{x}_i E\{\tilde{\mathbf{z}}_{ij}|\mathbf{x}_i\}^{\mathrm{T}}\right]\left[\sum_{i=1}^{N}h_{ij}E\{\tilde{\mathbf{z}}_{ij}\tilde{\mathbf{z}}_{ij}^{\mathrm{T}}|\mathbf{x}_i\}\right]^{-1},$$

$$\mathbf{\Psi}^{\text{new}} = \frac{1}{N}diag\left[\sum_{i=1}^{N}\sum_{j=1}^{M}h_{ij}(\mathbf{x}_i - \tilde{\mathbf{\Lambda}}_j^{\text{new}}E\{\tilde{\mathbf{z}}_{ij}|\mathbf{x}_i\})\mathbf{x}_i^{\mathrm{T}}\right], \quad (4)$$

where $\tilde{\mathbf{z}}_{ij} = [\mathbf{z}_{ij}\ 1]^{\mathrm{T}}$ is an augmented column vector of factors and $\tilde{\mathbf{\Lambda}}_j^{\text{new}} = [\mathbf{\Lambda}_j^{\text{new}}\boldsymbol{\mu}_j^{\text{new}}]$ is a new augmented factor loading matrix. In order to find more detailed derivations of above equations, see [13].

III. Continuous EDAs with MFAs

In the continuous optimization problems, candidate solutions are usually represented as real vectors. Most evolutionary algorithms for this problems maintain the population of these vectors to search for the optimal point. In this section, we describe how MFAs are applied to estimate the distribution of current population and to generate new offspring from the estimated distribution.

A. Estimation of Distribution by MFAs

MFAs can put similar individuals together in a group and estimate the density for each group simultaneously. This means that MFAs implicitly divide the current population into M sub-populations and find the values of latent variables and parameters to build a corresponding density model for each one by using the EM algorithm.

The parameters are initialized as follows[1]:

$$\pi_j^0 = 1/M,$$
$$\boldsymbol{\mu}_j^0 = \mathbf{N}_d\tilde{\mathbf{S}} + \boldsymbol{\mu},$$
$$\mathbf{\Lambda}_j^0 = \mathbf{N}_{d\times q}\sqrt{|\mathbf{S}|/q},$$
$$\mathbf{\Psi}^0 = diag[\mathbf{S}] + \delta\mathbf{I}, \quad (5)$$

[1]This initialization came from the Ghahramani's MFA software written in MATLAB. [Online] http://www.gatsby.ucl.ac.uk/~zoubin/software/mfa.tar.gz

where $\mathbf{S}$ is the sample covariance matrix, i.e., $\mathbf{S} = \sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}/N$, $\tilde{\mathbf{S}}$ is the principal square root of $\mathbf{S}$, i.e., $\tilde{\mathbf{S}}\tilde{\mathbf{S}} = \mathbf{S}$, $\mathbf{N}$ is a vector or matrix whose elements are randomly sampled from the standard normal distribution, and δ is a very small constant. In the E-step, the expectation values of unobserved variables are calculated by using equations (2) and (3). Then, in the M-step, we find the new parameter values which maximize the expected log-likelihood by using equation (4). These two steps are repeated until the proportional change of the log-likelihood is less then 10^{-4} or 100 steps of EM[2].

While conventional EDAs use the selected individuals from the current population to build the probability model, we estimate the density of the whole population by a MFA. This is because we adopt new selection method explained in the following section.

B. Generating New Population

To generate new individual $\mathbf{x}_i'$ corresponding to $\mathbf{x}_i$, we first have to determine which component density function is responsible for the data $\mathbf{x}_i$. According to the estimated posterior probability h_{ij}, $j = 1, \ldots, M$, the component j is selected. Then, we can sample easily the new individual $\mathbf{x}_i'$ from the conditional distribution of $\mathbf{x}_i'$ given the corresponding latent variable $\mathbf{z}_{ij}$ defined to be the Gaussian distribution $\mathcal{N}(\boldsymbol{\Lambda}_j\mathbf{z}_{ij} + \boldsymbol{\mu}_j, \boldsymbol{\Psi})$. This sampling task is trivial since the noise covariance matrix $\boldsymbol{\Psi}$ is diagonal.

In contrast to existing EDAs which deterministically replace the whole population or the worse part of the population by new individuals, our algorithm accepts the candidate solution according to the following probability

$$\begin{aligned}\alpha(\mathbf{x}_i, \mathbf{x}_i') &\equiv \min\left\{1, \frac{p(\mathbf{x}_i')}{p(\mathbf{x}_i)}\right\} \\ &= \min\left\{1, \frac{\exp\{-f(\mathbf{x}_i')\}}{\exp\{-f(\mathbf{x}_i)\}}\right\}, \qquad (6)\end{aligned}$$

where f is the fitness function for the given optimization problem. This acceptance rule says that the candidate individual is always accepted when the fitness of the candidate individual is better than that of the original one; otherwise, it is accepted according to the ratio of two probability. If the candidate solution is accepted, $\mathbf{x}_i'$ is copied into the next generation. If candidate is rejected, then, $\mathbf{x}_i$ is copied into the next generation. More theoretical foundation of this mechanism can be found in [14]. The whole procedure of our algorithm is summarized in figure 1.

[2]This termination condition is also from the Ghahramani's software

1. (**Initialize**) Randomly generate initial population whose size is N. Set generation count $g \leftarrow 0$.
2. (**MFA**) Start with the initialized parameters by the equation (5). Repeat until the stopping criterion for EM is met.
   - (E-step) Compute the expectation values of unobserved variables by using equations (2) and (3).
   - (M-step) Find the new parameter values that maximize the expected log-likelihood by using equation (4).
3. (**Generate**) Create N candidate solutions by choosing component j according to h_{ij} and then sampling data points from the Gaussian distribution $\mathcal{N}(\boldsymbol{\Lambda}_j\mathbf{z}_{ij} + \boldsymbol{\mu}_j, \boldsymbol{\Psi})$.
4. (**Select**) Construct the next population by selecting the solutions according to the equation (6).
5. (**Elitist**) Add the best individual of the previous generation to the next population.
6. (**Finish**) Stop if the termination criteria are met.
7. (**Loop**) Set $g \leftarrow g + 1$ and go to Step 2.

Fig. 1. Outline of the continuous estimation of distribution algorithms with MFA.

IV. Experimental Results

To verify the quality of our method, we use the Rosenbrock and Griewank functions,

$$f_{\text{Rosenbrock}}(\mathbf{x}) = \sum_{i=2}^{d}[100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2],$$

$$f_{\text{Griewank}}(\mathbf{x}) = 1 + \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right).$$

A. Comparative Results to Non-mixture Models

We compare the results of MFAs with that of single FAs to show the effect of mixture model in the continuous EDAs with latent variables. The population size N in all experiments was 1000 and the algorithm was stopped when the number of generation was 1000. Thus, all methods were given the same computational resource in the total number of function evaluations, i.e., 10^6.

Figure 2 illustrate the impact of EDAs with MFAs on the seven dimensional Rosenbrock's function optimization, where the range of all the components of the individual is $-2.048 \leq x_i \leq 2.048$. Without regard to the dimension of latent variables, the MFAs ($M \geq 2$) achieve better performance than the single FAs ($M = 1$). In all experiments by MFA methods, we found the satisfiable solutions except the two mixture model cases. The best one was 0.000293 at the point

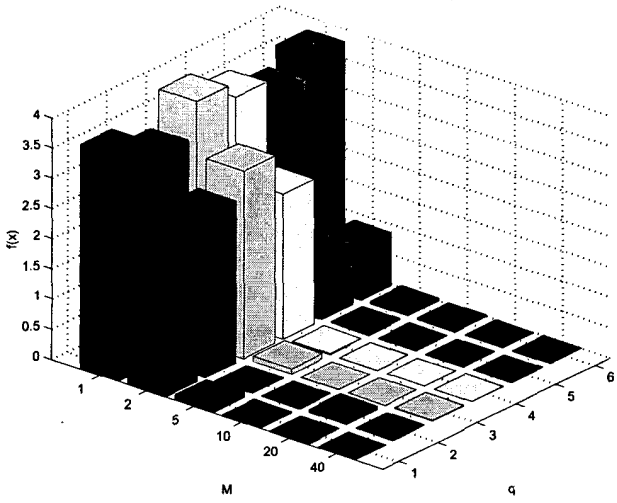

Fig. 2. Optimum values found in our experiments in terms of the number of mixtures and the dimension of the latent variables (minimum values among 10 runs for each setting).

$\mathbf{x} = (0.999750, 0.999493, 0.999283, 0.999912, 1.000168, 1.000931, 1.002600)$ when $M = 20$ and $q = 6$. This seems attributed to the fact that MFAs make the clusters of similar individuals which can be considered as sub-populations and appropriately estimate the density for each cluster. As the generation goes by, some of these clusters have the center points very close to the optimal and the diagonal elements of the error covariance matrix have small values. Therefore, we can search for the optimal solution easily by sampling new individuals from the corresponding distributions.

B. Analysis of Population Diversity

In the conventional EDAs, only some better individuals of the current population take part in the probability model building task. This may be helpful for an EDA to converge to a certain point, but it might reduce the population diversity drastically.

For the Rosenbrock's function optimization ($d = 7$), we compared our eMCMC-like selection method with the conventional style EDAs where the same MFAs ($M = 20$ and $q = 6$) were used as the density model, but only 50% (half) or 25% (quater) better individuals were selected to build the probability model and new individuals from this model replace the whole population. We can think that it is difficult to discover the diverse individuals in the population if the average fitness value is same as the best one. Thus the algorithm stopped when the difference

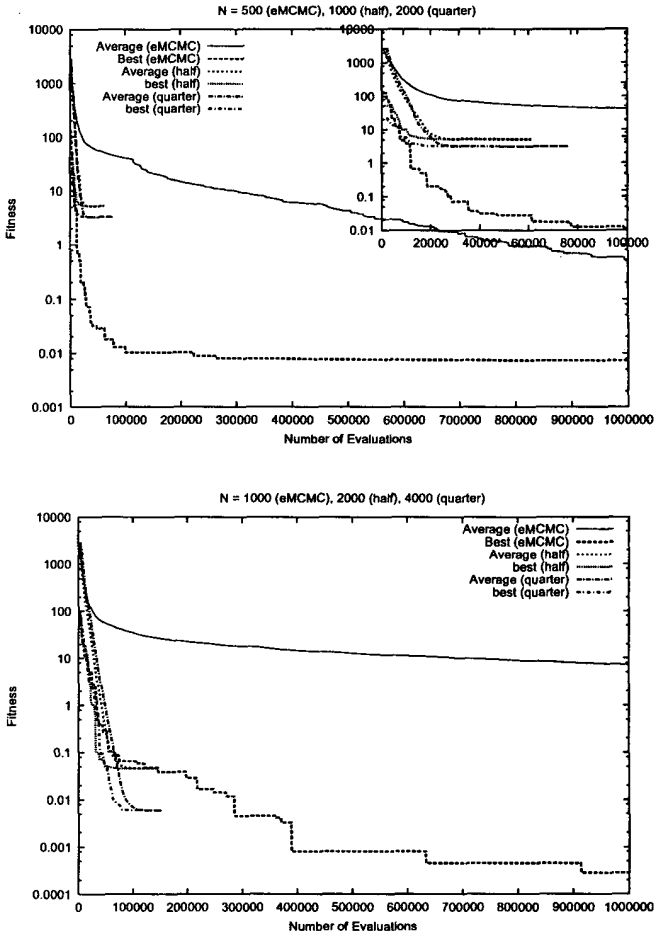

Fig. 3. Average and best fitness values for different selection methods (best case among 10 runs for each setting).

between the average and the best fitness is less than 10^{-5}.

Figure 3 shows the experimental results for two different population sizes. For the proper comparison, we used twice and four times individuals in the half and quarter case, respectively. While all conventional methods lost the population diversity in early stage and failed to obtain good solutions, our method maintained population diversity and also found much better solutions. This is because some worse candidate solutions can be included in the next population according the acceptance probability and it is helpful to escape from the local optima.

Table I shows the comparative results to other EDAs. Except the eMCMC selection method ($N = 1000$), all experiments use a population of 2000 individuals, from which the best 1000 solutions are selected to estimate the distribution. The MFA ($M = 10$, $q = 5$) with eMCMC selection have better performances than those of other EDAs and conventional style MFA.

TABLE I

MEAN FITNESS VALUES AND STANDARD DEVIATIONS AVERAGED ON 10 RUNS FOR THE TEN DIMENSIONAL ROSENBROCK AND GRIEWANK FUNCTIONS. (HERE, THE EARLIER REPORTED RESULTS CAME FROM [1].)

	Rosenbrock ($-10.0 \leq x_i \leq 10.0$)		Griewank ($-600.0 \leq x_i \leq 600.0$)	
Algorithm	Mean $\pm$ Stdev	# Evaluation	Mean $\pm$ Stdev	# Evaluation
$UMDA_c$	8.7204 ± 0.0382	301,850	$6.0783 \times 10^{-2} \pm 0.0193$	301,850
$MIMIC_c$	8.7141 ± 0.0164	301,850	$7.3994 \times 10^{-2} \pm 0.0286$	301,850
$EGNA_{BIC}$	8.8217 ± 0.16	268,067	$3.9271 \times 10^{-2} \pm 0.0243$	301,850
$EGNA_{BGe}$	8.6807 ± 0.0587	164,519	$7.6389 \times 10^{-2} \pm 0.0293$	301,850
$EGNA_{ee}$	8.7366 ± 0.0223	301,850	$5.6840 \times 10^{-2} \pm 0.0382$	301,850
MFA_{half}	8.7048 ± 2.5806	300,000	$7.7586 \times 10^{-3} \pm 0.0082$	300,000
MFA_{eMCMC}	2.5184 ± 1.2037	300,000	$1.0870 \times 10^{-3} \pm 0.0010$	300,000

V. Conclusions

We presented a new estimation of distribution algorithm based on the mixture of factor analyzers and stochastic selection. Our experimental results show that EDAs with MFAs are superior to EDAs with single FA and enough number of mixtures is required to solve hard optimization problems such as high dimensional Rosenbrock's function by EDAs based on the factor analysis. We also showed that stochastic selection methods play an important role in maintaining the population diversity to prevent the premature convergence into local optima. Thus we can conclude that these two schemes are essential for finding good solutions by EDAs with latent variables in the continuous optimization problems.

Acknowledgments

This research was supported in part by the Minister of Education and Human Resources Development under the BK21-IT Program. The RIACT at Seoul National University provides research facilities for this study.

References

[1] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2001.

[2] M. Pelikan, D. E. Goldberg, and F. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications*, vol. 21, no. 1, pp. 5-20, 2002.

[3] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Parallel Problem Solving from Nature - PPSN VI*, Lecture Notes in Computer Science, vol. 1917, pp. 385-394, Springer-Verlag, 2000.

[4] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, 2000.

[5] M. Gallagher, M. Frean, and T. Downs, "Real-valued evolutionary optimization using a flexible probability density estimator,", in *Proceedings of 1999 Genetic and Evolutionary Computation Conference*, vol. 1, pp. 840-846, Morgan Kaufmann Publishers, 1999.

[6] D. Thierens and P. Bosman, "Multi-objective optimization with iterated density estimation evolutionary algorithms using mixture models," in *Proceedings of the Third International Symposium on Adaptive Systems*, pp. 129-136, 2001.

[7] P. Bosman and D. Thierens, "Advancing continuous IDEAs with mixture distributions and factorization selection metrics," in *Proceedings of 2001 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 208-212, Morgan Kaufmann Publishers, 2001.

[8] R. Santana, A. Ochoa-Rodriguez, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proceedings of 2001 Genetic and Evolutionary Computation Conference*, pp. 543-550, Morgan Kaufmann Publishers, 1999.

[9] S.-Y. Shin, D.-Y. Cho, and B.-T. Zhang, "Function optimization with latent variable models," in *Proceedings of the Third International Symposium on Adaptive Systems*, pp. 145-152, 2001.

[10] C. M. Bishop, "Latent variable models," in *Learning in Graphical Models*, M. I. Jordan, Ed. pp. 371-403, The MIT Press, 1999.

[11] S.-Y. Shin and B.-T. Zhang, "Bayesian evolutionary algorithms for continuous function optimization," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 508-515, 2001.

[12] D.-Y. Cho and B.-T. Zhang, "Continuous estimation of distribution algorithms with probabilistic principal component analysis," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 521-526, 2001.

[13] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Technical Report CGG-TR-96-1, Department of Computer Science, University of Toronto, February 1997. [Online] http://www.gatsby.ucl.ac.uk/~zoubin/papers/tr-96-1.ps.gz

[14] B.-T. Zhang and D.-Y. Cho, "System identification using evolutionary Markov chain Monte Carlo," *Journal of Systems Architecture*, vol. 47, no. 7, pp. 587-599, 2001.

[15] D. J. Bartholomew and M. Knott, *Latent Variable Models and Factor Analysis*, 2nd ed. Arnold, 1999.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1-38, 1977.

[17] D. B. Rubin and D. T. Thayer, "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, no. 1, pp. 69-76, 1982.