

# Evolutionary Continuous Optimization by Distribution Estimation with Variational Bayesian Independent Component Analyzers Mixture Model

Dong-Yeon Cho and Byoung-Tak Zhang

Biointelligence Laboratory  
School of Computer Science and Engineering  
Seoul National University  
Seoul 151-742, Korea  
{dycho, btzhang}@bi.snu.ac.kr  
<http://bi.snu.ac.kr>

**Abstract.** In evolutionary continuous optimization by building and using probabilistic models, the multivariate Gaussian distribution and their variants or extensions such as the mixture of Gaussians have been used popularly. However, this Gaussian assumption is often violated in many real problems. In this paper, we propose a new continuous estimation of distribution algorithms (EDAs) with the variational Bayesian independent component analyzers mixture model (vbICA-MM) for allowing any distribution to be modeled. We examine how this sophisticated density estimation technique has influence on the performance of the optimization by employing the same selection and population alternation schemes used in the previous EDAs. Our experimental results support that the presented EDAs achieve better performance than previous EDAs with ICA and Gaussian mixture- or kernel-based approaches.

## 1 Introduction

In the variation operators, particularly mutations of the continuous evolutionary algorithms such as evolution strategy, probability distributions have been used to generate new offspring. Although one of the most popular distributions is Gaussian in continuous cases, we have no prior knowledge of which type of the probability distributions is suitable for the given problem and how the parameter values of the distributions are determined to guide further searches toward the optimal point. This is the reason why we study the effective methods to estimate the probability distributions of the good solutions in the current population. For the last decade, many researchers developed this kind of evolutionary optimization techniques called estimation of distribution algorithms (EDAs) [1] for both discrete and continuous cases. Detailed reviews of existing EDAs can be found in [2] and [3].

For the continuous cases, most of the previous EDAs are based on the multivariate Gaussian distribution and their variants or extensions although there

are a few exceptions such as marginal histogram models [4]. In spite of the wide usage of the Gaussian distribution, we suffer from the size of covariance matrix as the dimensionality of the problem increases. This can be alleviated by ignoring the covariance structure (i.e., employing diagonal covariance matrix). In fact, this is assumed in the simplest version of EDAs. However, it extremely cuts down the flexibility of the model.

The conditional factorization of a multivariate joint probability with conditional dependence assumptions can reduce the number of required parameters to define the probability distribution. It can also explicitly represent the conditional dependencies among the variables of the problem by an acyclic directed graph. Larrañaga *et al.* [5] used a probabilistic graphical model called Gaussian network which can be considered as a factorization tool. It decomposes the probability distribution of the given problem into various factors or conditional probabilities. Bosman and Thierens [6] present continuous IDEEA instances which also exploit the fact that every multivariate joint probability distribution can be written as a conditional factorization.

We can readily implement this factorization by introducing latent or hidden variables. Latent variable model provides a powerful approach to probabilistic model-building by supplementing a set of directly observed variables with additional latent variables [7]. 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 space. In addition, it is easy to sample new data from the estimated distribution since latent variable models are generative. More comprehensive explanation and experimental results for our continuous EDAs based on latent variable models can be found in [8] and [9].

Since the data often have the intricate structure which is difficult to capture by a high-order-dependency model such as the Gaussian network, we should consider more complicated probability models. Clustering techniques can be an useful solution to alleviate these difficulties. After making some groups of similar data, we are able to use the existing probability models to obtain an appropriate interpretation for each cluster. Mixture models provide a natural way to handle these clusters in EDAs. They can also build very complex distributions through a proper choice of its components to represent accurately the local areas of support of the true distribution. For continuous cases, the Gaussian mixture models have been successfully applied to the function optimization in [10] and [11]. We also presented a mixture version of continuous EDAs with latent variable models which are corresponding to Gaussian mixtures [12].

As an extreme case of the mixture models, Gaussian Kernels are used in the IDEEA frameworks [13]. The localized aspect of the kernel method is better than that of the normal distribution since there is a density function for each data point. However, it completely misses the linkage information and tends to quickly overfit the data. Therefore, the performance of the optimization is seriously affected by the density of sample points and kernel variances. To overcome these

difficulties, the mixed Bayesian optimization algorithm (MBOA) uses a Bayesian network with decision trees whose leaf nodes are one-dimensional Gaussian kernel estimators [14]. MBOA is an extension of hierarchical Bayesian Optimization Algorithm (hBOA) [15] from binary to continuous domain. Actually, MBOA can solve both the discrete and continuous problems, but we concentrate on the continuous cases in this paper.

One of the drawbacks in these Gaussian mixture- or kernel-based approaches is that each component or separated domain should be Gaussian. However, this cannot hold in many real cases. In this paper, we propose a new continuous EDAs with the variational Bayesian independent component analyzers mixture model (vbICA-MM) [16]. It can not only model non-Gaussian, separated clusters by incorporating a very flexible ICA model, but also automatically determine the local dimensionality of each cluster by employing recent developments in variational Bayesian inference. We examine how this sophisticated density estimation technique has influence on the performance of the optimization by using the same population alternation schemes as the mixture version of iterated density estimation evolutionary algorithm (MIDEA) and MBOA.

The paper is organized as follows. In Section 2, the basic concept of vbICA-MM is explained. Section 3 presents the EDAs with vbICA-MM for the continuous domain. Then, Section 4 reports the experimental results and their analysis on some benchmark functions. Finally, conclusions of this study are drawn in Section 5.

## 2 Theoretical Background for vbICA-MM

### 2.1 Independent Component Analysis Model

Independent component analysis (ICA) tries to model an observed  $S$ -dimensional data vector  $\mathbf{x}$  as a linear combination of statistically independent sources (latent variables)  $\mathbf{s}$  of dimension  $L$  with added Gaussian noise

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{y} + \mathbf{e},$$

where  $\mathbf{y}$  is an  $S$ -dimensional bias vector,  $\mathbf{A}$  is the  $S \times L$  mixing matrix, and  $\mathbf{e}$  is  $S$ -dimensional additive noise which is assumed to be zero-mean Gaussian and isotropic with precision  $\lambda I$ . Then, the conditional probability of observing a data vector  $\mathbf{x}^n$  is given by

$$p(\mathbf{x}^n | \mathbf{A}, \mathbf{s}^n, \mathbf{y}, \lambda) = \left( \frac{\lambda}{2\pi} \right)^{\frac{S}{2}} \exp \left\{ -\frac{\lambda}{2} (\mathbf{x}^n - \mathbf{A}\mathbf{s} - \mathbf{y})^T (\mathbf{x}^n - \mathbf{A}\mathbf{s} - \mathbf{y}) \right\}.$$

Since the sources  $\mathbf{s} = \{s_1, s_2, \dots, s_L\}$  are mutually independent by definition and a mixture of only 1-dimensional Gaussians with  $m_i$  components per source is adopted, the distribution over  $\mathbf{s}$  for data point  $n$  can be written as

$$p(\mathbf{s}^n | \varphi) = \prod_{i=1}^L \sum_{q_i=1}^{m_i} \pi_{i,q_i} \mathcal{N}(s_i^n; \mu_{i,q_i}, \beta_{i,q_i}),$$

where  $\mu_{i,q_i}$  and  $\beta_{i,q_i}$  are the mean and precision of Gaussian  $q_i$  in source  $i$  respectively. The mixture proportions  $\pi_{i,q_i}$  are the prior probabilities of choosing component  $q_i$  of the  $i$ th source. The parameters of source  $i$  are  $\varphi_i = \{\pi_i, \mu_i, \beta_i\}$  and the complete parameter set of the source model is  $\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_L\}$ . This mixture of Gaussian source model allows any distribution to be modeled [17].

The complete collection of possible source states is denoted  $\mathbf{q} = \{q_1, q_2, \dots, q_m\}$  and runs over all  $m = \prod_i m_i$  possible combinations of source states. By integrating and summing over the hidden variables,  $\{\mathbf{s}, \mathbf{q}\}$ , the likelihood of the IID data  $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$  given the model parameters  $\Theta = \{\mathbf{A}, \mathbf{y}, \lambda, \varphi\}$  can now be written as

$$p(\mathbf{X}|\Theta) = \prod_{n=1}^N \sum_{\mathbf{q}=1}^m \int p(\mathbf{x}^n, \mathbf{s}^n, \mathbf{q}^n|\Theta) d\mathbf{s}, \quad (1)$$

where  $d\mathbf{s} = \prod_i ds_i$ .

## 2.2 ICA Mixture Model

We suppose that a data vector  $\mathbf{x}^n$  is generated from a  $C$ -component mixture model given assumptions  $\mathcal{M}$ . Then, the probability can be written in the form

$$p(\mathbf{x}^n|\mathcal{M}) = \sum_{c=1}^C p(c|\mathcal{M}_0) p(\mathbf{x}^n|\mathcal{M}_c, c), \quad (2)$$

where  $\mathcal{M} = \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_C\}$  is the vector of assumptions about the mixture process,  $\mathcal{M}_0$ , and component model assumptions,  $\mathcal{M}_c$ . The variable  $c$  indicates which component of the mixture model is chosen to generate a given data vector.

For the ICA mixture model, we specify a form for  $p(c|\mathcal{M}_0)$  and substitute (1) into (2). Then, we can quantify the likelihood of the observed data under the ICA mixture model as follows

$$p(\mathbf{x}^n|\mathcal{M}) = \sum_{c=1}^C p(c|\kappa) p(\mathbf{X}|\Theta_c, c),$$

where  $p(c|\kappa) = \{p(c=1) = \kappa_1, p(c=2) = \kappa_2, \dots, p(c=C) = \kappa_C\}$  and  $\Theta_c$  is the parameters for the  $c$ th ICA model. We can obtain the maximum likelihood (ML) estimation for these parameters as well as the values of the latent variables by using gradient descent or the EM algorithm. However, the ML methods can easily get caught in local maxima and fail to determine the best model structure since there is no consideration of model complexity.

## 2.3 Variational Bayesian Learning for vbICA-MM

Bayesian approaches overcome the weakness of ML methods by integrating out the parameters  $\{\kappa, \Theta_c\}$  and hidden variables  $\{\mathbf{s}_c, \mathbf{q}_c\}$  and penalizing more complex models which can lead to overfitting. For this purpose, the prior distributions over the model parameters and hidden variables are properly chosen to be

conjugate to allow the tractability. A detailed statement for the priors over the ICA mixture models can be found in [16].

Although Bayesian inference has the ability to handle the overfitting and select the best model structure, it is often computationally intensive and analytically intractable in practice. One of the useful tools for Bayesian integration is the variational approximation. We consider the log evidence for data  $\mathbf{X}$  and form a lower bound on it using Jensen's inequality:

$$\log p(\mathbf{X}) = \log \int d\mathbf{W} p(\mathbf{X}, \mathbf{W}) \geq \int d\mathbf{W} p'(\mathbf{W}) \log \frac{p(\mathbf{X}, \mathbf{W})}{p'(\mathbf{W})} \equiv F[\mathbf{W}], \quad (3)$$

where  $\mathbf{W}$  is the vector of all hidden variables and unknown parameters, and  $p'(\mathbf{W})$  is a tractable approximation to the posterior  $p(\mathbf{W}|\mathbf{X})$ . We can compare various models by calculating  $F$  for each model as well as implicitly integrate out the unknowns  $\mathbf{W}$  by maximizing  $F$ .

For the tractable Bayesian learning of the ICA mixture model, the following factorization of the distribution of the parameters and hidden variables is used:

$$p'(\mathbf{W}) = p'(\mathbf{c})p'(s_c|\mathbf{q}_c, c)p'(\mathbf{q}_c|c)p'(\boldsymbol{\kappa})p'(\mathbf{y})p'(\boldsymbol{\lambda})p'(\mathbf{A})p'(\boldsymbol{\alpha})p'(\boldsymbol{\varphi}), \quad (4)$$

where  $p'(\boldsymbol{\varphi}) = p'(\boldsymbol{\pi})p'(\boldsymbol{\mu})p'(\boldsymbol{\beta})$ ,  $p'(a|b)$  is the approximating density of  $p'(a|b, \mathbf{X})$ , and  $\boldsymbol{\alpha}$  is the precision vector for each column of mixing matrix  $\mathbf{A}$  whose elements have a zero-mean Gaussian as the prior distribution. The posteriors over the sources also factorize such that

$$p'(s_c|\mathbf{q}_c, c) = \prod_{i=1}^{L_c} p'(q_i|c)p'(s_{c,i}|q_i, c).$$

This additional factorization allows efficient scaling of computation with the number of latent variables. By substituting  $p(\mathbf{X}, \mathbf{W})$  and (4) into (3), we obtain expressions for the bound,  $F$ , to the ICA mixture model. In order to know how the measure  $F$  is maximized, that is, how the vbICA-MM algorithm is implemented in detail, see [16].

### 3 Continuous EDAs with vbICA-MM

#### 3.1 Distribution Estimation by vbICA-MM

Most evolutionary algorithms for continuous optimization problems maintain a population of real vectors to search for an optimal point. The vbICA-MM can put similar individuals together in a group and estimate the density for each group simultaneously. This means that vbICA-MM implicitly divides the current population into  $C$  sub-populations and finds the values of latent variables and parameters to build a corresponding density model for each sub-population by variational Bayesian learning. Unlike the mixture of Gaussian or other Gaussian-based latent variable models, each cluster can have a non-Gaussian distribution

in the vbICA-MM. This is because the source model can enclose the distributions with positive and negative kurtosis and complex multimodal distributions.

We first select the good candidates from the current population as the data used in the density estimation step. Both selection schemes of MIDEA and MBOA were tried in our experiments. MIDEA adopts the truncation selection where the best  $\lfloor \tau N \rfloor$  ( $0 < \tau < 1$ ) vectors are taken. In MBOA,  $\lfloor \tau N \rfloor$  vectors are chosen by repeating the tournament selection (where the tournament size is two).

To estimate the probability distribution of the selected population, we use the MATLAB vbICA-MM software from Choudrey<sup>1</sup>. Although we can determine which model is preferred, that is, what is the best value for  $C$  by monitoring  $F$  according to the number of mixtures, we just fix the value of  $C$  in our experiments for the fast estimation. We also utilize the maximum number of sources, which means that the source vector has the same dimensions as the data ( $L = S$ ). However, the relevance of each source may be automatically determined. Column  $i$  of the mixing matrix  $\mathbf{A}_c$  will be close to zero if the precision  $\alpha_i$  for the column is large, indicating source  $i$  is irrelevant. The ICA mixture model is trained until  $F$  changed by less than 0.01 or the number of iterations reached 10.

### 3.2 Generating a New Population

To create a new individual  $\tilde{\mathbf{x}}^n$  from the trained model, we first have to determine which component density  $c$  is responsible for the randomly chosen parent  $\mathbf{x}^n$  from the selected vectors as the data. According to a probability proportional to the estimated component posterior probability, the component  $c$  is selected. Then, we can sample easily the new individual  $\tilde{\mathbf{x}}^n$  from the conditional distribution given the corresponding latent variable  $\mathbf{s}_c^n$  defined to be the Gaussian distribution  $\mathcal{N}(\mathbf{A}_c \mathbf{s}_c^n + \mathbf{y}_c, \lambda_c I)$ . This sampling task is trivial since we can have the reconstructed values of hidden variables and parameters from the trained model and the noise is assumed to be zero-mean Gaussian and isotropic. As mentioned earlier, however, the source distribution is not a simple Gaussian but the factorized mixture of Gaussians, which makes the vbICA-MM more flexible than Gaussian mixtures and the mixture version of other latent variable models. This sampling procedure is repeated until  $(N - \lfloor \tau N \rfloor)$  vectors are obtained.

Besides the way to estimate the density, a population alternation method is also important. We tried both schemes of MIDEA and MBOA with vbICA-MM. The new sampled vectors replace the worst  $(N - \lfloor \tau N \rfloor)$  vectors in MIDEA. To ensure an effective niching, MBOA employs restricted tournament replacement (RTR) proposed originally in [18]. RTR selects a subset of the original population for each new sampled offspring. The size of these subsets is fixed to some constant, called the window size. Then, the new offspring competes with the most similar member of the subset. If the new one is better, it replaces the corresponding individual; otherwise, the new one is discarded.

<sup>1</sup> [online] [http://www.robots.ox.ac.uk/~riz/Code/vbICA1\\_0.tar.gz](http://www.robots.ox.ac.uk/~riz/Code/vbICA1_0.tar.gz)

## 4 Experimental Results

To evaluate the quality of the proposed EDA with vbICA-MM, we compare our EDAs with three evolutionary optimization algorithms based on the learning of probability distribution. For fair experiments, each selection and population alternation schemes of three previous algorithms is incorporated into the EDA with vbICA-MM, respectively.

### 4.1 UMDA/ICA

As a first attempt to combine EDAs with ICA, Zhang *et al.* [19] proposed univariate marginal distribution algorithm with independent component analysis (UMDA/ICA). This algorithm applies ICA to the points in the selected parent set before the density is estimated. Since this preprocessing via ICA eliminates some nonlinear interrelations among the variables, we can prevent a bad approximation of the density by using UMDA [1] which assumes all variables are independent.

To compare our EDAs with UMDA/ICA, we used same test function as [19]:

$$f(x_1, x_2, x_3, x_4) = \sum_{i=1}^3 \sum_{j=i+1}^4 \{100(x_i^2 - x_j)^2 + (1 - x_1)^2\}, \quad |x_i| \leq 2.048.$$

In this function, each variable interacts with every other variable. The smallest function value is 0 at point (1, 1, 1, 1). The population size  $N$  in all experiments were 3,000. All algorithms were stopped if the number of generation is equal to 120. We used 2-tournament selection like UMDA/ICA. The population is divided into 1,500 groups of 2 points, and the point with lower function value in each group is selected. All selected points are directly entered into the next generation and the half of the population is filled with the new sampled points from the estimated probability density. For vbICA-MM, we set three components per source mixture of Gaussian and the number of mixtures  $C$  for data vectors is 4. As shown in Table 1, our EDAs with vbICA-MM has a better performance and stability than UMDA/ICA.

**Table 1.** Comparative results with UMDA/ICA. The objective function values are averaged over 10 independent runs. Here, the results for UMDA and UMDA/ICA came from [19]. (<sup>†</sup>We have no idea about what the precision is.)

Algorithm	Objective function value		
	Mean $\pm$ Stdev	Best	Worst
UMDA	4.6247 $\pm$ 0.537	3.528	5.189
UMDA/ICA	2.499 $\pm$ 1.650	0 <sup>†</sup>	4.704
EDA with vbICA-MM	$4.316 \times 10^{-6} \pm 2.617 \times 10^{-6}$	$1.002 \times 10^{-6}$	$9.740 \times 10^{-6}$

## 4.2 MIDEA

To compare our EDAs with MIDEA, we use two benchmark functions. One of them is a ten-dimensional Rosenbrock function,

$$f_{Ro}(\mathbf{x}) = \sum_{i=2}^{10} (100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2).$$

Although this function is unimodal, there are non-linear dependencies between variables. It has parabolic valleys along the curves  $x_i = x_{i-1}^2$  in the search interval  $|x_i| \leq 5.12$  with the minimum 0 at point  $(1, \dots, 1)$ . The other is a ten-dimensional Fletcher-Powell function,

$$f_{FP}(\mathbf{x}) = \sum_{i=1}^{10} \left( \sum_{j=1}^{10} (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j) - \sum_{j=1}^{10} (a_{ij} \sin x_j + b_{ij} \cos x_j) \right)^2,$$

where  $a_{ij}$  and  $b_{ij}$  are random integers in the range  $[-100, 100]$ . Obviously the minimum value is 0 at  $x_j = \alpha_j \in [-\pi, \pi]$ . This function is highly multimodal and not symmetric. We used the same values for  $a_{ij}$ ,  $b_{ij}$  and  $\alpha_j$  published in [20].

The parameter setting for MIDEA is as follow. The maximum amount of clusters is 8 and the threshold of the BEND leader algorithm used for clustering in MIDEA is 0.3 (which is determined to make 8 clusters in almost every generation). We also allowed the maximum number of parents for any variable in the Gaussian networks. The portion  $\tau$  of the selected individuals to build the probabilistic model is 0.3. To choose the best population size for MIDEA, we used the testing sequence  $N = [200, 400, 800, 1600, 3200, 6400]$  performing 20 runs for each size. From these tests, we obtained the best average fitness when  $N$  is 6400 for both functions. Same parameter values are applied to our EDAs with vbICA-MM. All algorithms are terminated when the number of function evaluation is  $10^6$ .

Table 2 shows the comparative results. Although we fail to obtain the optimum value of the Rosenbrock function for both methods, our EDAs with vbICA-MM has a little better performance than MIDEA. However, our methods outperform MIDEA for the Fletcher-Powell function while consuming more time than MIDEA.

## 4.3 Fletcher-Powell Function

With the above testing sequence, we set the population size for MBOA. The best mean fitness values are shown when  $N$  is 3200 for Rosenbrock function and 800 for Fletcher-Powell function. These population sizes are also used in our methods. The window size in RTR is  $N/20$  and the number of mixtures  $C$  for vbICA-MM is 8. All algorithms are allowed  $10^6$  function evaluations.

For Rosenbrock function, there is no significant difference of the performance. However, this function can be optimized more efficiently by EDAs with vbICA-MM than MBOA. Our EDAs are better than MBOA for Fletcher-Powell function with respect to the performance and stability while our methods are more time-consuming than MBOA for this multimodal case (Table 3).

**Table 2.** Comparative results with MIDEA. Mean fitness values with standard deviations and relative time (RT) averaged on 20 runs.

Algorithm	Rosenbrock		Fletcher-Powell	
	Mean $\pm$ Stdev	RT	Mean $\pm$ Stdev	RT
MIDEA	$7.538 \pm 0.049$	6.888	$103.861 \pm 31.404$	0.255
EDA with vbICA-MM	$3.954 \pm 1.501$	51.302	$6.417 \times 10^{-3} \pm 0.010$	47.284

**Table 3.** Comparative results with MBOA. Mean fitness values with standard deviations and relative time (RT) averaged on 20 runs.

Algorithm	Rosenbrock		Fletcher-Powell	
	Mean $\pm$ Stdev	RT	Mean $\pm$ Stdev	RT
MBOA	$2.886 \times 10^{-2} \pm 0.0314$	5135.15	$4.243 \times 10^{-1} \pm 1.272$	7.573
EDA with vbICA-MM	$1.693 \times 10^{-2} \pm 0.0307$	500.53	$2.320 \times 10^{-2} \pm 0.007$	378.65

## 5 Conclusions

We presented a new estimation of distribution algorithm based on the variational Bayesian independent component analyzers mixture model. The influence of this novel density estimation technique on the optimization performance is analyzed by adopting the same selection and population alternation schemes used in the previous EDAs. Experimental results show that EDAs with vbICA-MM outperform the ICA-combined UMDA. They are also better than EDAs with Gaussian mixture- or kernel-based approaches for two benchmark functions. This superiority is more evident in the multimodal case than unimodal one.

## Acknowledgments

This research was supported by the Ministry of Commerce, Industry and Energy through the MEC project, the Ministry of Science and Technology through National Research Lab (NRL), and the Ministry of Education and Human Resources Development under the BK21-IT Program. The ICT at Seoul National University provided research facilities for this study.

## References

1. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distribution I. Binary Parameters. In: Voigt, H.-M. *et al.* (eds.): *Parallel Problem Solving from Nature IV*. Lecture Notes in Computer Science, Vol. 1141. Springer (1996) 178-187
2. Larrañaga, P.: A Review on Estimation of Distribution Algorithms. In: Larrañaga, P., Lozano, J.A. (eds.), *Estimation of Distribution Algorithms: A New Tools for Evolutionary Computation*. Kluwer Academic Publishers (2001) 57-100

3. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A Survey of Optimization by Building and Using Probabilistic Models. *Computational Optimization and Applications* **21** (2002) 5-20
4. Tsutsui, S., Pelikan, M., Goldberg, D.E.: Evolutionary Algorithm using Marginal Histogram in Continuous Domain. In *Proceedings of the GECCO-2001 Workshop Program*. (2001) 230-233
5. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks. In *Proceedings of the GECCO-2000 Workshop Program*. (2000) 201-204
6. Bosman, P., Thierens, D.: Expanding from Discrete to Continuous Estimation of Distribution Algorithms: The IDEA. In: Schoenauer, M. *et al.* (eds.): *Parallel Problem Solving from Nature VI*. Lecture Notes in Computer Science, Vol. 1917. Springer (2000) 767-776
7. Bishop, C.M.: Latent Variable Models. In: Jordan, M.I. (ed.), *Learning in Graphical Models*. The MIT Press (1999) 371-403
8. Shin, S.-Y., Zhang, B.-T.: Bayesian Evolutionary Algorithms for Continuous Function Optimization. In *Proceedings of 2001 Congress on Evolutionary Computation*, Vol. 1. (2001) 508-515
9. Cho, D.-Y., Zhang, B.-T.: Continuous Estimation of Distribution Algorithms with Probabilistic Principal Component Analysis. In *Proceedings of 2001 Congress on Evolutionary Computation*, Vol. 1. (2001) 521-526
10. Gallagher, M., Freaun, M., Downs, T.: Real-Valued Evolutionary Optimization Using a Flexible Probability Density Estimator. In: *Proceedings of 1999 Genetic and Evolutionary Computation Conference*, Vol. 1. (1999) 840-846
11. Bosman, P., Thierens, D.: Advancing Continuous IDEAs with Mixture Distributions and Factorization Selection Metrics. In *Proceedings of the GECCO-2001 Workshop Program*. (2001) 208-212
12. Cho, D.-Y., Zhang, B.-T.: Evolutionary Optimization by Distribution Estimation with Mixtures of Factor Analyzers. In *Proceedings of 2002 Congress on Evolutionary Computation*, Vol. 2. (2002) 1396-1401
13. Bosman, P., Thierens, D.: IDEAs Based on the Normal Kernels Probability Density Function. Technical Report, UU-CS-2000-11. Utrecht University (2002)
14. Ocenasek, J., Schwarz, J.: Estimation of Distribution Algorithm for Mixed Continuous-Discrete Optimization. In *Proceedings of the 2nd Euro-International Symposium on Computational Intelligence*. (2002) 227-232
15. Pelikan, M., Goldberg, D.E.: Escaping Hierarchical Traps with Competent Genetic Algorithms. In: Spector, L. *et al.* (eds.): *Proceedings of 2001 Genetic and Evolutionary Computation Conference*. (2001) 511-518
16. Choudrey, R.A., Roberts, S.J.: Variational Mixture of Bayesian Independent Component Analyzers. *Neural Computation* **15** (2003) 213-252
17. Attias, H.: Independent Factor Analysis. *Neural Computation* **11** (1999) 803-851
18. Harik, G.R.: Finding Multimodal Solutions Using Restricted Tournament Selection. In: Eshelman, L.J. (ed.): *Proceedings of the sixth International Conference on Genetic Algorithms*. (1995) 24-31
19. Zhang, Q., Allinson, N.M., Yin, H.: Population Optimization Algorithm Based on ICA. In: Yao, X. and Fogel, D. B. (eds.): *Proceedings of the First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*. (2000) 33-36
20. Bäck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press (1996)