Continuous Estimation of Distribution Algorithms with Probabilistic Principal Component Analysis

Dong-Yeon Cho
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
dycho@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@scai.snu.ac.kr

Abstract- Recently, many evolutionary algorithms have been studied to build and use an probability distribution model of the population for optimization problems. Most of these methods tried to represent explicitly the relationship between variables in the problem with factorization techniques or the graphical model such as Bayesian or Gaussian network. Thus enormous computational cost is required for constructing those models when the problem size is large. In this paper, we propose new estimation of distribution algorithm by using probabilistic principal component analysis (PPCA) which can explains the high order interactions with the latent variables. Since there are no explicit search procedures for the probability density structure, it is possible to rapidly estimate the distribution and readily sample the new individuals from it. Our experimental results support that presented estimation of distribution algorithms with PPCA can find good solutions more efficiently than other EDAs for the continuous spaces.

1 Introduction

Many evolutionary algorithms have been proposed that model the probability distributions of the good solutions and use these probabilistic models to generate new populations. That is, there are neither crossover nor mutation operators and instead the new individuals are sampled from the probability distribution. These methods are generally called the estimation of distribution algorithms or EDAs [11].

One of the main issue in this field is how to estimate the accurate distribution that can capture the structure of the given problem. The simplest way for distribution estimation is to assume that each variable in a problem is independent. Population-based incremental learning (PBIL) [1], univariate marginal distribution algorithm (UMDA) [10], and compact genetic algorithm (cGA) [7] for the discrete space and PBILc [15] for the continuous space belong to this class. However they are not appropriate for learning any interdependencies between variables. To capture the pairwise dependencies, mutual information maximizing input clustering (MIMIC) [6], dependency tree algorithm [2], and bivariate marginal distribution algorithm (BMDA) [13] were proposed. Chain, tree, and forest structures were used in each method, respectively.

Covering some pairwise interactions is still insufficient to solve problems with high-order dependencies. To capture more complex dependencies, Mühlenbein and Mahnig [12] present the factorized distribution algorithm (FDA). Here, the distribution is decomposed into various factors or conditional probabilities, and then this factorized distribution is used as a fixed model. FDA can be extended to an algorithm, LFDA, which computes a good factorization from the data with Bayesian networks. Pelikan et al. [14] propose the Bayesian Optimization Algorithm (BOA) which uses the techniques for modeling multivariate data by Bayesian networks in order to estimate the distribution of promising solutions.

To search good probability density models in continuous spaces, integrated density estimation evolutionary algorithm (IDEA) [4] used the Kullback-Leibler divergence as a distance metric to the full joint probability density structure and tested the various probability density functions for each element in the probability density structure. Larrañaga et al. [9] replace the Bayesian network with the Gaussian network for the continuous domain and employed four score metrics to construct the networks for the selected individuals.

All these methods except the simplest ones use distribution models to represent explicitly the relationship between variables in the problem. However, the problem of determining the best model with respect to a given score metric is usually very hard. For example, finding the optimal Bayesian network for a given dataset is NP-complete [5]. That is, enormous time is required for building the models when the problem size is large. Thus most researchers adopted greedy version of the original algorithm to prevent this ill-behavior although the exact distributions cannot be estimated by using the realistic methods. Recently, Zhang and Shin [17] developed another type of EDA, where the Helmholtz machines are used to model and sample from the distribution of selected individuals without explicit expression of multivariate interactions. They empirically showed that the learning time tends to grow linearly as the problem complexity or size increase.

In this paper, we propose new estimation of distribution algorithm with probabilistic principal component analysis (PPCA) [16] which can also cover higher order interactions with latent variables like the Helmholtz machines. Since there is no explicit search procedure for the probability density struc-

0-7803-6657-3/01/$10.00 ©2001 IEEE

ture, it is possible to rapidly estimate the distribution and easily sample the new individuals from it. This method can also be applied to the discrete space, however we focus on the continuous domain for demonstration purposes.

The paper is organized as follows. In section 2 we explain the basic concept of PPCA. Section 3 presents the EDA with PPCA algorithms for the continuous domain. Section 4 reports the results of experiments for some benchmark functions and Section 5 summarizes our findings in this study.

2 Basic Concept of Probabilistic Principal Component Analysis

2.1 Principal Component Analysis

Principal component analysis (PCA) is a powerful technique in data analysis. The central idea of PCA is to reduce the dimensionality of a data set which consists of many interrelated variables. It is achieved by searching for the direction in data-space which have the highest variance, and subsequently projecting the data onto it [8].

For a set of observed d dimensional data vector $\{\mathbf{x}_i\}$, $i \in \{1, 2, ..., N\}$, we define the q principal axes $\{\mathbf{w}_j\}$, $j \in \{1, 2, ..., q\}$ as those orthonormal axes onto which the retained variance under projection is maximal. Then it can be shown that the vector $\{\mathbf{w}_j\}$ are given by the q dominant eigenvectors which correspond to the largest eigenvalues of the sample covariance matrix

$$\mathbf{S} = \frac{1}{N}\sum_{i}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}, \tag{1}$$

where $\boldsymbol{\mu}$ is the data sample mean,

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i}^{N}\mathbf{x}_i, \tag{2}$$

such that $\mathbf{S}\mathbf{w}_j = \lambda_j \mathbf{w}_j$. The q principal components of the observed data $\mathbf{x}_i$ are given by the vector $\mathbf{z}_i = \mathbf{W}^{\mathrm{T}}(\mathbf{x}_i - \boldsymbol{\mu})$, where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_q)$. The variables z_j are uncorrelated such that the covariance matrix $\sum_i \mathbf{z}_i \mathbf{z}_i^{\mathrm{T}}/N$ is diagonal with elements λ_j. A complementary property of PCA is that the principal component projection of all orthogonal linear projections minimizes the squared reconstruction error $\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$, where the optimal linear reconstruction of $\mathbf{x}_i$ is given by $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}$.

However, PCA is not a probabilistic model thus Tipping and Bishop [16] addressed this limitation by using the latent variable model which is closely related to factor analysis.

2.2 Factor Analysis

A latent variable model tries to relate a d dimensional data $\mathbf{x}$ to a corresponding q dimensional latent variables $\mathbf{z}$. In standard factor analysis [3], the relationship is linear:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{3}$$

where the latent variables $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ have a unit isotropic Gaussian, the noise model is Gaussian $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Psi})$ with diagonal covariance matrix $\boldsymbol{\Psi}$, and $\mathbf{z}$ is independent of $\boldsymbol{\epsilon}$. From this formulation, the data $\mathbf{x}$ has also Gaussian distribution $\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \boldsymbol{\Psi})$.

The key assumption for this model is that the observed variables x_i are conditionally independent given the values of the latent variables $\mathbf{z}$ because of the diagonality of $\boldsymbol{\Psi}$. Thus these latent variables are intended to explain the correlations between observation variables while $\boldsymbol{\epsilon}$ represents the independent noise.

2.3 Probabilistic principal component analysis

For the isotropic Gaussian noise model $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, equation (3) implies that $\mathbf{z}$ conditional probability distribution over $\mathbf{x}$-space is given by $\mathbf{x}|\mathbf{z} \sim N(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$, i.e,

$$p(\mathbf{x}|\mathbf{z}) = (2\pi\sigma^2)^{-d/2} \exp\Big\{ - \frac{\|\mathbf{x} - \mathbf{W}\mathbf{z} - \boldsymbol{\mu}\|^2}{2\sigma^2}\Big\}. \tag{4}$$

With the marginal distribution of the latent variables $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ defined by

$$p(\mathbf{z}) = (2\pi)^{-q/2} \exp\Big\{ - \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{z}\Big\}, \tag{5}$$

the marginal distribution for the observed data $\mathbf{x}$ is obtained by integrating out the latent variables as follows:

$$\begin{aligned} p(\mathbf{x}) &= \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \\ &= (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\Big\{ - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\Big\}, \end{aligned} \tag{6}$$

where the covariance is specified by $\boldsymbol{\Sigma} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$ and this implies $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The posterior distribution of $\mathbf{z}$ is easily obtained by standard methods and it also turns out to be normal. That is, $\mathbf{z}|\mathbf{x} \sim N(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}), (\sigma^{-2}\mathbf{W}^{\mathrm{T}}\mathbf{W} + \mathbf{I})^{-1})$. Thus the posterior distribution of the latent variables $\mathbf{z}$ given the observed $\mathbf{x}$ can be calculated:

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) = (2\pi)^{-q/2}|\sigma^{-2}\mathbf{C}|^{1/2} \\ \times \exp\Big[- \frac{1}{2}\Big\{\mathbf{z} - \mathbf{C}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu})\Big\}^{\mathrm{T}}(\sigma^{-2}\mathbf{C}) \\ \Big\{\mathbf{z} - \mathbf{C}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu})\Big\}\Big], \end{aligned} \tag{7}$$

where $\mathbf{C} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$. The mean of this distribution might then be used to predict $\mathbf{z}$ for a given $\mathbf{x}$ and the precision of the predictions would be given by the elements of the covariance matrix.

Although there is no closed form analytic solution for $\mathbf{W}$ and σ, the parameters for this model can be obtained by iterative procedure, e.g. by using expectation-maximization (EM) algorithms which will be explained in the next section.

3 Continuous Estimation of Distribution Algorithms with PPCA

In the continuous optimization problems, candidate solutions are usually represented as real vectors. PPCA can explain the relationship between each component of promising solution vectors with the latent variables. The procedures used in PPCA to obtain the values of the variables and to generate new instances with those values are described in this section.

3.1 Estimation of Distribution by PPCA

The selected N individuals among the current population whose size is M are regarded as the observed data and the EM approach to maximizing the likelihood for PPCA is employed. Through this procedure, the distribution of the data points can be estimated.

For a selected individual $\mathbf{x}_i$, the value of corresponding $\mathbf{z}_i$ is unknown. However the joint distribution $p(\mathbf{x}, \mathbf{z})$ of the given samples and latent variables is known, thus we can calculate the expectation of the log-likelihood. In the E-step of the EM algorithm, the expectation with respect to the posterior distribution of $\mathbf{z}_i$ given the selected $\mathbf{x}_i$ is computed. In the M-step, new parameter values of $\mathbf{W}$ and σ^2 are determined that maximize the expected log-likelihood.

Using equations (4) and (5), the log-likelihood is defined as follows:

$$\begin{aligned} L &= \sum_{i=1}^{N} \ln\{p(\mathbf{x}_i, \mathbf{z}_i)\} \\ &= \sum_{i=1}^{N} \ln\Big[(2\pi\sigma^2)^{-d/2} \exp\Big\{-\frac{\|\mathbf{x}_i - \mathbf{W}\mathbf{z}_i - \boldsymbol{\mu}\|^2}{2\sigma^2}\Big\} \\ &\qquad \times (2\pi)^{-q/2} \exp\Big\{-\frac{1}{2}\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_i\Big\}\Big]. \end{aligned} \tag{8}$$

In the E-step, we take the expectation of L with respect to the distribution $p(\mathbf{x}_i|\mathbf{z}_i, \mathbf{W}, \sigma^2)$:

$$\begin{aligned} E\{L\} = -\sum_{i=1}^{N} \Big[&\frac{d}{2}\ln\sigma^2 + \frac{1}{2}\mathrm{tr}\big(E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\}\big) \\ &+ \frac{\|\mathbf{x}_i - \boldsymbol{\mu}\|^2}{2\sigma^2} - \frac{1}{\sigma^2}E\{\mathbf{z}_i\}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}(\mathbf{x}_i - \boldsymbol{\mu}) \\ &+ \frac{1}{2\sigma^2}\mathrm{tr}\big(\mathbf{W}^{\mathrm{T}}\mathbf{W}E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\}\big)\Big], \end{aligned} \tag{9}$$

where we have omitted terms independent of the model parameters and

$$E\{\mathbf{z}_i\} = \mathbf{C}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x}_i - \boldsymbol{\mu}), \tag{10}$$

$$E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\} = \sigma^2\mathbf{C}^{-1} + E\{\mathbf{z}\}E\{\mathbf{z}\}^{\mathrm{T}}, \tag{11}$$

with $\mathbf{C} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$. Note that these statistics are computed using the current values of the parameters and follow from distribution (7).

1. (**Initialize**) Randomly generate initial population whose size is M. Set generation count $g \leftarrow 0$.
2. (**Selection**) Select N promising solutions.
3. (**PPCA**) Start with randomly initialized parameters and the sample mean given by the equation (2).
   - (E-step) Compute the expectation value of the latent variables and their covariances by using equations (10) and (11).
   - (M-step) Find the parameters that maximize the expected log-likelihood by using equations (12) and (13).

   Repeat until the stopping criterion for EM is met.
4. (**Generate**) Create new population by sampling M/N data points for each latent variable from the Gaussian distribution $N \sim (\mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \sigma^2\mathbf{I})$.
5. (**Finish**) Stop if the termination criteria are met.
6. (**Elitist**) Add the best individual of the previous generation to the generated population.
7. (**Loop**) Set $g \leftarrow g + 1$ and go to Step 2.

Figure 1: Outline of the continuous estimation of distribution algorithms with PPCA

In the M-step, the expectation of log-likelihood $E\{L\}$ is maximized with respect to $\mathbf{W}$ and σ^2 by differentiating equation (9) and setting the derivatives to zero. This gives the new parameter estimates

$$\mathbf{W}_{\mathrm{new}} = \Big[\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})E\{\mathbf{z}_i\}^{\mathrm{T}}\Big]\Big[\sum_{i=1}^{N}E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\}\Big]^{-1}, \tag{12}$$

$$\begin{aligned} \sigma^2_{\mathrm{new}} = \frac{1}{Nd}\sum_{i=1}^{N}\Big[&\|\mathbf{x}_i - \boldsymbol{\mu}\|^2 - 2E\{\mathbf{z}_i\}^{\mathrm{T}}\mathbf{W}_{\mathrm{new}}(\mathbf{x}_i - \boldsymbol{\mu}) \\ &+ \mathrm{tr}\big(E\{\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\}\mathbf{W}_{\mathrm{new}}^{\mathrm{T}}\mathbf{W}_{\mathrm{new}}\big)\Big]. \end{aligned} \tag{13}$$

To maximize the likelihood, the sufficient statistics of the conditional distributions are calculated form the E-step equations (10) and (11), and revised estimates for the parameters are obtained from the M-step equations (12) and (13). These four equations are repeated sequentially until convergence is achieved or some other stopping criterion is met.

3.2 Generating New Population

The conditional distribution of the new individual $\mathbf{x}'_i$ given the corresponding latent variable $\mathbf{z}_i$ is defined to be Gaussian, $N \sim (\mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \sigma^2\mathbf{I})$, like equation (4). To generate next population, thus, we only have to sample M/N data points for each latent variable $\mathbf{z}_i$ from the Gaussian distribu-

tion with parameters obtained by the previous EM algorithm. The whole procedure is summarized in Figure 1.

4 Experiments

4.1 Test functions

Various test functions have been used in order to compare our method with others.

- Ackley's function:

$$f_{Ack}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e,$$
$$-30.0 \le x_i \le 30.0$$

- Rastrigin's function:

$$f_{Ras}(\mathbf{x}) = 10d + \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i))$$
$$-5.12 \le x_i \le 5.12$$

- Test function 1:

$$f_1(\mathbf{x}) = \left\{10^{-5} + \sum_{i=1}^{d}|y_i|\right\}^{-1},$$
$$y_1 = x_1;\ y_i = y_{i-1} + x_i, i = 2, ..., d;$$
$$-0.16 \le x_i \le 0.16$$

- Test function 2:

$$f_2(\mathbf{x}) = \sum_{i=1}^{d}[(x_1 - x_i^2)^2 + (x_i - 1)^2];$$
$$-10.0 \le x_i \le 10.0$$

- Test function 3:

$$f_3(\mathbf{x}) = 1 + \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right),$$
$$-600.0 \le x_i \le 600.0$$

- Test function 4:

$$f_4(\mathbf{x}) = 100 f_1(\mathbf{x}),$$
$$-3.0 \le x_i \le 3.0$$

- Test function 5:

$$f_5(\mathbf{x}) = 100\left\{10^{-5} + \sum_{i=1}^{d}|y_i|\right\}^{-1},$$
$$y_1 = x_1;\ y_i = \sin(y_{i-1}) + x_i, i = 2, ..., d;$$
$$-3.0 \le x_i \le 3.0$$

- Test function 6:

$$f_6(\mathbf{x}) = 100\left\{10^{-5} + \sum_{i=1}^{d}|y_i|\right\}^{-1},$$
$$y_i = 0.024(i+1) - x_i, i = 1, ..., d;$$
$$-3.0 \le x_i \le 3.0$$

Their characteristics and parameter settings used in the experiments are shown in Table 1.

4.2 Results of the Experiments

In figures 2 and 3, the results for Ackley's function and Rastrigin function are presented. The number of function evaluations and total running time in our experiments are measured on the Pentium II 400MHz PC with 128MB memory. Algorithms are stopped when the best fitness value is below 1.0×10^{-15}. Note that the total running time as well as the number of function evaluations is increasing linearly as the problem size grows.

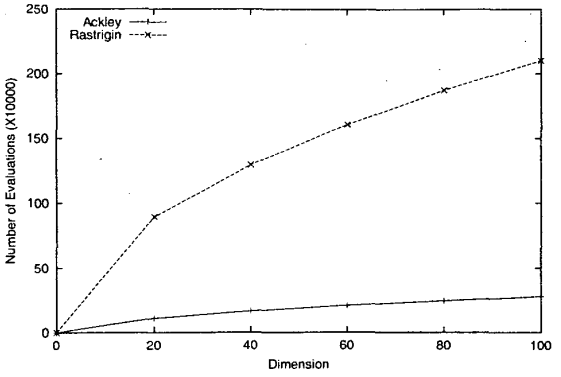

Figure 2: Computational cost in terms of the number of the function evaluations (averaged over 10 runs).

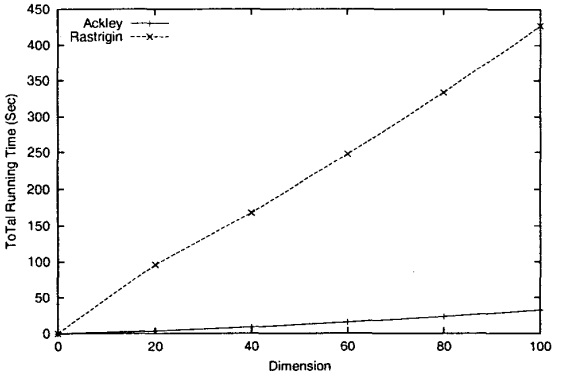

Figure 3: Computational cost in terms of the total running time (averaged over 10 runs).

Table 1: Characteristics of the functions and parameter settings used in the experiments

Function	Ackley	Rastrigin	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Dimension	20-100	20-100	10	10	10	100	100	100
#Max_Eval	-	-	300,000	300,000	300,000	200,000	200,000	200,000
Type	Min	Min	Max	Min	Min	Max	Max	Max
Optimum	0	0	10^5	0	0	10^7	10^7	10^7
Pop_Size	1,000	10,000	10,000	2,000	2,000	2,000	2,000	800
Selection Rate	0.1	0.1	0.01	0.1	0.1	0.1	0.2	0.1
#Latent_Variables	1	1	5	1	1	10	20	1

Table 2: Best fitness values averaged on 100 runs for the test function 1, 2, and 3. Total running time for the PPCA-based approach is also given.

Methods	$UMDA_c^G$	$MIMIC_c^G$	$EGNA_{ee}$	$EGNA_{BGe}$	ES	PPCA	
Functions	Mean	Mean	Mean	Mean	Mean	Mean	Total Time
Test 1	53460	58775	100000	100000	5910	94063.01	25.34
Test 2	0.13754	0.13397	0.09914	0.0250	0	3.68×10^{-9}	6.01
Test 3	0.011076	0.007794	0.008175	0.012605	0.034477	0	5.71

Table 3: Best fitness values averaged on 20 runs for the test function 4, 5, and 6 with the standard deviation. Relative time of IDEA and PPCA is also given.

Methods	(10+50)-ES	$PBIL_c$	IDEA		PPCA		
Functions	Mean ± Stdev	Mean ± Stdev	Mean	RT	Mean ± Stdev	RT	Total Time
Test 4	2.91 ± 0.45	4.76 ± 0.78	7.50	44.32	7.83 ± 0.75	6.91	102.05
Test 5	7.56 ± 1.52	11.18 ± 1.36	27.73	4.95	18.23 ± 1.54	2.34	769.05
Test 6	399.07 ± 6.97	4803 ± 4986	9999999.96	130.17	9997761.50 ± 1161.42	15.86	19.80

Table 2 displays the results for test functions 1, 2, and 3. Results obtained by all other methods except PPCA are taken from [9] in which Gaussian networks are used as the probabilistic model. One hundred experiments for each function and algorithm were carried out.

The PPCA-based approach outperformed the methods based on the Gaussian networks which can represent the identical class of distributions as the UMDA and MIMIC. This implies that PPCA can detect the high order relations between variables by using the latent variables (or just one variable). Our algorithm provided solutions which are very close to the real optimum values for the test function 2 and found the optimum value for the function 3 within about 6 seconds, while it could not find the optimal points on the fitness landscape presented by the test function 1.

Table 3 summarizes the best fitness values averaged over 20 runs with standard deviations for the test functions 4, 5, and 6. Here, the earlier reported results came from [15] and [4]. All algorithms end far from the actual optimum for the test functions 4 and 5, however PPCA also have better performances than those of the standard ES and the continuous version of PBIL.

The performance of PPCA is better than that of IDEA for the test function 4 and comparable to for the function 5, although it is worse for the function 5. However, note that the relative time (RT) spent by PPCA is far smaller than that of IDEA for all functions. RT is was introduced from [4] as a CPU independent fair running time comparison metric. Therefore, we can expect that the performance of PPCA will be improved by using a little more computational cost.

For all experiments, the number of latent variables for all problems was determined empirically. Even if many problems can be solved with the small number of latent variable, it is difficult to determined the numbers for hard problems which have highly correlated relations between variables such as test functions 4 and 5. A constructive algorithm which starts from one latent variable and increase the number of those variables during learning can be adopted to find the optimal numbers, thus finally to have better answers. This is one of our future works.

5 Conclusions

We presented a estimation of distribution algorithm that is based on the probabilistic principal component analysis. Our empirical results show that EDAs with PPCA is superior to simple algorithms which can not detect the multivariate interactions and is also highly competitive to other complex distribution estimation algorithms. One of the advantages of our algorithms is that the computational cost tends to increase lin-

early as the problem complexity grows. Thus it can find good solutions more efficiently than other EDAs for the continuous space.

Acknowledgements

This research was supported in part by the Korea Ministry of Science and Technology through KISTEP under grant BR-2-1-G-06 and by the BK21-IT Program.

Bibliography

[1] Baluja, S. and Caruana, R. (1995) "Removing the genetics from the standard genetic algorithm," *Proceedings of the 12th International Conference on Machine Learning*, pp. 38-46, Morgan Kaufmann.

[2] Baluja, S. and Scott, D. (1997) "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," *Proceedings of the 14th International Conference on Machine Learning*, pp. 30-38, Morgan Kaufmann.

[3] Bartholomew, D.J. and Knott, M. (1999) *Latent Variable Models and Factor Analysis*, 2nd Edition, Arnold.

[4] Bosman, P.A.N. and Thierens, D. (2000) "Expanding from discrete to continuous estimation of distribution algorithms: The IDEA," *Lecture Notes in Computer Science 1917: Parallel Problem Solving from Nature - PPSN VI*, pp. 767-776, Springer.

[5] Chickering, D.M. (1996) "Learning Bayesian networks is NP-Complete," *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121-130, Springer.

[6] De Bonet, J.S., Isbell, C.L., and Viola, P. (1997) "MIMIC: Finding optima by estimating probability densities," *Advances in Neural Information Processing Systems*, vol. 9, pp. 424-430, The MIT Press.

[7] Harik, G.R., Lobo, F.G., and Goldberg, D.E. (1998) "The compact genetic algorithm," *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pp. 523-528.

[8] Jolliffe, I.T. (1986) *Principal Component Analysis*, Springer.

[9] Larrañaga, P., Etxeberria, R., Lozano, J.A., and Peña, M. (2000) "Optimization in continuous domains by learning and simulation of Gaussian networks," *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 201-204.

[10] Mühlenbein, H. and Paaß, G. (1996) "From recombination of genes to the estimation of distributions I. Binary parameters," *Lecture Notes in Computer Science 1141: Parallel Problem Solving from Nature - PPSN IV*, pp. 178-187, Springer.

[11] Mühlenbein, H., Mahnig, T., and Ochoa Rodriguez, A. (1999) "Schemata, distributions and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, no. 2, pp. 215-247.

[12] Mühlenbein, H. and Mahnig, T. (1999) "FDA - A scalable evolutionary algorithms for the optimization of additively decomposed functions," *Evolutionary Computation*, vol. 7, no. 4, pp. 353-376.

[13] Pelikan, M. and Mühlenbein, H. (1999) "The bivariate marginal distribution algorithm," *Advances in Soft Computing - Engineering Design and Manufacturing*, pp. 521-535, Springer.

[14] Pelikan, M., Goldberg, D.E., and Cantú-Paz, E. (2000) "Linkage Problem, Distribution Estimation, and Bayesian Networks," *Evolutionary Computation*, vol. 8, no. 3, pp. 311-340.

[15] Sebag, M. and Ducoulombier, A. (1998) "Extending population-based incremental learning to continuous search spaces," *Lecture Notes in Computer Science 1498: Parallel Problem Solving from Nature - PPSN V*, pp. 418-427, Springer.

[16] Tipping, M.E. and Bishop, C.M. (1999) "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B*, vol. 61, no. 3, pp. 611-622.

[17] Zhang, B.-T. and Shin, S.-Y. (2000) "Bayesian evolutionary optimization using Helmholtz machines" *Lecture Notes in Computer Science 1917: Parallel Problem Solving from Nature - PPSN VI*, pp. 827-836, Springer.