2009 International Conference of Soft Computing and Pattern Recognition

A Novel Fuzzy Histogram based Estimation of Distribution Algorithm for Global Numerical Optimization

Weili Liu, Jing-hui Zhong,wei-gang Wu,Jing Xiao and Jun Zhang (Corresponding author)
Dept. of Computer Science,SUN Yat-sen University, Guangzhou, P.R.China
e-mail: junzhang@ieee.org

Abstract—**Applying Estimation of Distribution Algorithms (EDAs) to solve continuous problems is a significant and challenging task in the field of evolutionary computation. So far, various continuous EDAs have been developed based on different probability models. Initially, the EDAs based on a single Gaussian probability model are widely used but they have trouble in solving multimodal problems. Later EDAs based on a mixture model and on a clustering technique are then introduced to conquer such drawback. However, they are either time consuming or need prior knowledge of the problems. Recently, the histogram has begun to be used in continuous EDAs, but the histogram based EDAs (HEDAs) usually need too much time and space to gain a highly accurate solution. On the basis of pioneering contributions, this paper proposes a fuzzy histogram based EDA (FHEDA) for continuous optimization. In the FHEDA, the estimated range of the fuzzy histogram is adjusted adaptively by the current promising solutions, which leads the algorithm to search good solutions efficiently. A mutation mechanism is also introduced in the sampling operation to avoid being trapped in local optima. The performance of the proposed FHEDA is evaluated by testing seven benchmark functions with different characteristics. Two Gaussian based EDAs and the sur-shr-HEDA are studied for comparison. The results show that among all experimental algorithms, the FHEDA can give comparatively satisfying performance on unimodal and multimodal functions.**

Keywords- Estimation of Distribution Algorithms, Fuzzy, Histogram, Numerical Optimization, Evolutionary Algorithms

I. INTRODUCTION

Estimation of Distribution Algorithms (EDAs), first introduced by Baluja [1] and H.Muhlenbein [2], are a novel class of population based evolutionary algorithms (EAs). The general procedures of the EDAs are quite similar to those of the Genetic Algorithms (GAs). However, there is neither crossover nor mutation in the EDAs. Instead, an EDA generates new offspring by sampling from a probability model, which is estimated from some current promising solutions. As the EDAs have few parameters to set and have good capabilities of solving optimization problems, great progresses have been made to the EDAs in recent years and many complex problems in real-world have been solved by them.

The EDAs were first introduced to solve discrete optimization problems with binary representations [1, 2]. Like many other EAs [3]-[6], the research emphasis on the EDAs has gone through a transformation from the discrete field to the continuous field. The first attempt to apply an EDA to solve continuous problems was made by Servet [4]. From then on, various methods have been proposed to enhance the performance of continuous EDAs [5]-[13]. According to the probability model used, there are two main categories of continuous EDAs. The first category focuses on the Gaussian probability model and the second category concentrates on the histogram probability model.

Early continuous EDAs in the first category assume that the distribution of the promising solutions accords with a single Gaussian probability model. However, this impractical assumption makes these algorithms easily get trapped in local optima when solving multimodal problems. It is because that a single Gaussian probability model is far from enough to estimate the distribution of promising solutions, which tend to form various groups. In order to do away with this shortcoming, several efforts have been made. For example, Bosnian et al. [7] used the Gaussian Mixture Model (GMM) to estimate the distribution of promising solutions. However, the learning task of the GMM was time consuming. For another example, Lu et al. [8] introduced a Rival Penalized Competitive Learning (RPCL) clustering technique to detect the number of global optima. Though the algorithms worked well for unimodal problems and simple multimodal problems with a few local optima, the maximum number of clusters must be defined beforehand, which was infeasible when dealing with multimodal problems with a great deal of local optima.

The first histogram based EDA was proposed by Tsutsui et al. [9]. After that, Yuan Bo et al. proposed a histogram based PBIL ($\mathrm{PBIL_H}$) [10], which utilized the fitness values of the population when constructing a histogram. Ding Nan et al. [11] enhanced the performance of the HEDA by considering both the historical and current information of the population. Mutation and elitist strategies were also introduced in [11]. Though the histogram model has advantages in capturing several local optima, it has some trouble in applications. One is that a highly accurate solution can only be achieved by setting a large number of bins, which needs heavy computational cost. To conquer such drawback, Q.F.Zhang [12] proposed a Histogram based EDA with a uniform design method and two local search algorithms (EDA/L). The key idea behind the EDA/L is to use a histogram probability model for coarse-grained search and to use two different local search algorithms (cheap and expensive) for fine-grained search. Ding Nan et al. [13] proposed a sur-shr-HEDA, which utilized the surrounding effect of individuals and the shrink strategies.

Unlike previous HEDAs, this paper proposes a novel Fuzzy Histogram based EDA (FHEDA) for continuous

978-0-7695-3879-2/09 $26.00 © 2009 IEEE
DOI 10.1109/SoCPaR.2009.30

optimization. Since originally introduced by Zadeh in 1965, the notion of the fuzzy sets has achieved great success both in theoretical researches and industrial applications [14]-[17]. Based on the theory of fuzzy sets, K.loquin et al. [18] have introduced the fuzzy histogram density estimator and provided its feature analysis in depth. Their work inspires us to use the fuzzy histogram instead of the crisp histogram in the EDA, so as to gain a truer and more stable density estimation of the promising solutions. There are several features of the FHEDA. For one thing, the estimated range of the fuzzy histogram is adjusted adaptively by analyzing current promising solutions, which leads the algorithm to search highly accurate solutions. Specifically, we first find the minimum and maximum promising solutions (here only considered one dimension problems for convenience), and then extend the range between them slightly to be the estimated range of the fuzzy histogram. This mechanism enables the FHEDA to shrink or extend the estimated range of the fuzzy histogram appropriately. For another thing, by defining the number of bins and a fuzzy partition of the estimated range, we can compute the height associated to each bin by checking each solution one by one efficiently. For each solution, it contributes to the heights of two neighbor bins at the same time but with different weights. Therefore, the final height associated to every bin in the fuzzy histogram is smoother than that of the crisp histogram, which makes the algorithm perform more stably. Moreover, a simple mutation mechanism is also introduced in the sampling operation to avoid trapping into local optima. The performance of the proposed FHEDA is evaluated by testing seven benchmark functions with different characteristics. Two Gaussian based EDAs and the recently published sur-shr-HEDA are studied for comparison.

The rest of this paper is organized as follows. Section II briefly describes the general procedures of the EDAs and their features. Section III illustrates the implementations of the FHEDA in details. The experimental studies on the FHEDA are presented in Section IV. At last, Section V draws the conclusions.

II. Brief Review on the EDAs

A. Framework of the EDAs

Procedures of the EDAs are generally similar to those of the GAs. However, the EDAs have neither crossover nor mutation, but they include a probability model building operation and a sampling operation to generate new offspring. The common outline of the EDAs is as follows.

1) Initialization

In this process, parameters of the probability model, the population size *Popsize* and the number of promising solutions *Se* are initialized. To form the initial population, *Popsize* individuals are generated randomly in the search space and their fitness values are evaluated.

2) Selection of individuals

This process selects *Se* individuals from the current population. These selected individuals are used to construct the probability model in the following operations.

3) Construct probability model

A probability model is built in this step, based on the statistical information extracted from the *Se* selected individuals.

4) Generate new offspring

A set of new offspring are generated according to the previously constructed probability model and their fitness values are evaluated.

There is a repetition from Step2 to Step4 to evolve the population and the evolution does not stop until some termination criteria are met. The termination criteria depend on specific needs, for example, reaching a predefined maximum number of generations or finding a preset fitness value for the best individual and so on.

B. Classification of the EDAs

The core of an EDA is its probability model, which has significant influence on the behaviors of the algorithm. Since the first proposition of the EDAs, various probability models have been introduced. From the viewpoint of the capability to capture interactions between variables, the EDAs can be classified into three following categories.

1) No interactions: In this class of the EDAs, probability models are constructed by assuming that the variables in the problems are independent. Early EDAs, including the PBIL [1] and the UMDA [2] belong to this class.

2) Pairwise interactions: This group of the EDAs takes the interaction between two variables into account. Famous EDAs of this class are the MIMIC [19], the COMIT [20] and so on.

3) Multivariate interactions: This class usually uses complex probability models that can cover multivariate interactions. The EDAs of this sort, like the BOA [21], are usually applied to solve complex problems.

III. Fuzzy Histogram Based EDA

A. Fuzzy Histogram Description

Given $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ as a set of data points within the estimated range $[x_{\min}, x_{\max}]$ and N as the number of bins, the histogram of $\boldsymbol{X}$ can be defined to be the vector $\boldsymbol{H} = \{h_1, h_2, \ldots, h_N\}$, where h_i is the number of data points falling into ith bin ($i = 1, 2, \ldots, N$), as computed by

$$h_i = \sum_{j=1}^{n} I_i(x_j) \ , i = 1, 2, \ldots, N \quad , \tag{1}$$

where $I_i(x)$ is the characteristic function of bin i defined as (2), and Δx is the bandwidth of each bin, which can be computed by (3).

$$I_i(x) = \begin{cases} 1 \ , & \text{if } (i\text{-}1) \le \dfrac{x - x_{\min}}{\Delta x} < i \\ 1 \ , & \text{if } x = x_{\max} \text{ and } i = N \\ 0 \ , & \text{otherwise} \end{cases} \tag{2}$$

$$\Delta x = (x_{\max} - x_{\min}) / N \tag{3}$$

With the histogram defined above, an estimated density $\hat{f}_H(x)$ of the underlying probability density of x □ $\boldsymbol{X}$ is given by

$$\hat{f}_H(x) = \frac{\sum_{i=1}^{N}(I_i(x)\cdot h_i)}{n\cdot\Delta x}. \tag{4}$$

Due to its simplicity and effectiveness, the histogram is widely used to estimate the probability density of data points. However, the histogram density estimator has some weaknesses, one of which is the excessive effects of the estimated range and the bandwidth on the estimated density [18]. In recent years, several researchers suggest using the fuzzy histogram to overcome this drawback of the crisp histogram.

The fuzzy histogram is constructed based on a fuzzy partition of the estimated range. Given a set of fixed points $\boldsymbol{S}$ = {s_1, s_2, …, s_N} within the estimated range $\Omega = [x_{\min}, x_{\max}]$ with $x_{\min} = s_1 < s_2 < \ldots < s_N = x_{\max}$, we can define N fuzzy subsets A_1, A_2, …, A_N, with membership functions $\mu_{A_1}(x), \mu_{A_2}(x), ..., \mu_{A_N}(x)$, to form a fuzzy partition of Ω. Conditions on these N fuzzy subsets are presented in [18].The membership function $\mu_{A_i} : \boldsymbol{R} \to [0,1], i = 1,...,b$ can be triangular, cosine or others.

By introducing the concept of the fuzzy partition, we can extend the definition of the crisp histogram to the fuzzy histogram: Given $\boldsymbol{X}$ = {x_1, x_2, …, x_n} as a data set within the estimated range [$x_{\min}$, $x_{\max}$] and N as the number of bins, the fuzzy histogram of $\boldsymbol{X}$ is defined as the vector $\boldsymbol{H}$ = {h_1, h_2, …, h_N}, where h_i, (i = 1, 2, …, N) is computed by

$$h_i = \sum_{j=1}^{n}\mu_{A_i}(x_j),\ \ i = 1,2,...,N,\ \ i = 1, 2, \ldots, N. \tag{5}$$

After constructing the fuzzy histogram, we can estimate the underlying probability density of x □ $\boldsymbol{X}$ by

$$\hat{f}_{FH}(x) = \frac{\sum_{i=1}^{N}(h_i\cdot\mu_{A_i}(x))}{n\cdot(\frac{x_{\max}-x_{\min}}{N-1})}. \tag{6}$$

Further studies in [18] show that the estimated results of the fuzzy histogram are truer than that of the crisp histogram as well as less sensitive to the choices of the number of bins.

B. Algorithm Framework

The framework of the FHEDA is shown in Fig. 1, and the implemental details of the FHEDA are accordingly described as follows.

1) Initialization

This step is responsible to initialize several parameters, including the number of bins *BIN_NUM*, the rate of mutation q_0, the population size *Popsize* and the number of promising solutions *Se*. Besides, the initial population is generated randomly in the whole search space and their fitness values are evaluated.

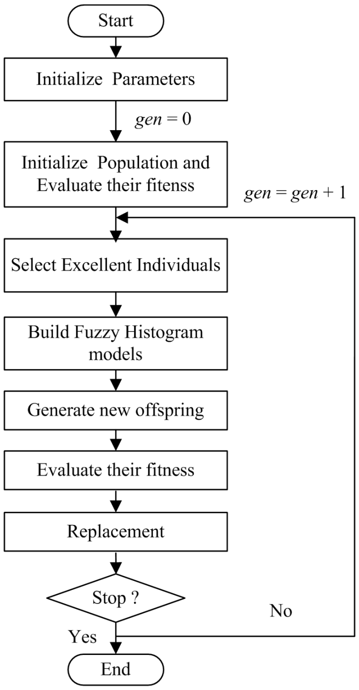

Figure 1. Flowchart of the FHEDA

2) Selection of excellent individuals

This step selects *Se* excellent individuals to construct the probability model used in the following operations.

3) Building histogram models

In this step, a set of histogram models are constructed by extracting the information from the *Se* selected individuals. The number of histogram models depends on the dimensions of the problem. For a problem of M dimensions, the solution can be represented as $\boldsymbol{X} = (x_1, x_2, ..., x_M)$. In order to estimate the distribution of *Se* solutions, M histogram models are needed and each model corresponds to a variable x_i. Take a one-dimension problem for example, let $\boldsymbol{P}$ = {p_1, p_2, …, p_{Se}} be *Se* better individuals and the procedures of building a fuzzy histogram are as follows.

Step 1: Set the estimated range of the histogram model. In this study, the estimated range of the histogram model [*Lbound*, *Ubound*] corresponding to $\boldsymbol{P}$ is compute by

$$Lbound = \begin{cases} p_{\min} - \varepsilon\cdot\Delta x & \text{if } p_{\min} - \varepsilon\cdot\Delta x \geq LBOUND \\ LBOUND & \text{otherwise} \end{cases} \tag{7}$$

and

$$Ubound = \begin{cases} p_{\max} + \varepsilon\cdot\Delta x & \text{if } p_{\max} + \varepsilon\cdot\Delta x \leq UBOUND \\ UBOUND & \text{otherwise} \end{cases}, \tag{8}$$

where $p_{\min}$ and $p_{\max}$ are the minimum and maximum value in

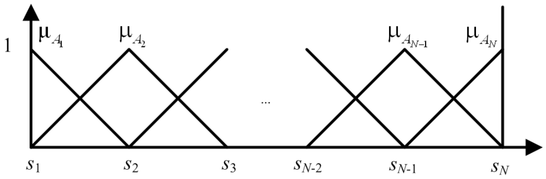

Figure 2. Member functions of N fuzzy subsets

P respectively, *LBOUND* and *UBOUND* are the defined minimum and maximum value of ***P***, ε is the extending rate and Δx is the distance value defined as (9).

Step 2: Construct fuzzy partition of the search space. With a predefined value N, we can gain N fixed points $\boldsymbol{S} = \{s_1, s_2, \ldots, s_N\}$ on the interval $\Omega = [Lbound, Ubound]$ with $x_{\min} = s_1 < s_2 < \ldots < s_N = x_{\max}$ The distance between each point is the same, which is

$$\Delta x = (Ubound - Lbound)/(N-1). \tag{9}$$

With these N fixed points, we can define N fuzzy subsets A_1, A_2, …, A_N on Ω . In this paper, the membership functions of these N fuzzy subsets are shown as Fig. 2. Notice that, μ_{A_1} and μ_{A_N} are different from other membership functions, they cover a data interval that is only half the size of the others'.

Step 3: Compute the height associated to each fuzzy subsets h_i by (5).

4) Generating new offspring

After building the histogram models, *Popsize* new offspring are then sampled from the models. In order to bring diversities to the new population, we add a mutation mechanism. Specifically, let $\boldsymbol{X} = (x_1, x_2, ..., x_M)$ be a solution, and each x_i can be generated by applying the rule given by

$$x_i = \begin{cases} v & \text{, if } q \le q_0 \text{ (exploitation)} \\ R(t) & \text{, otherwise (biased exploration)} \end{cases}, \tag{10}$$

where q_0 is a parameter with value within [0, 1], q is a random value uniformly distributed between 0 and 1, v is a random value uniformly distributed in the whole search space, and $R(t)$ returns a value generated by the histogram model of x_i in generation t. Note that by using a crisp interpolation method, we can get the upper bounds and the lower bounds of the selected fuzzy subset by (11) and (12).

$$l_k = \begin{cases} Lbound & , \text{ if } k = 1 \\ Ubound - \Delta x/2 & , \text{ if } k = N \\ Lbound + (k-1)\cdot\Delta x - \Delta x/2 & , \text{ otherwise} \end{cases} \tag{11}$$

$$u_k = \begin{cases} Lbound + \Delta x/2 & , \text{ if } k = 1 \\ Ubound & , \text{ if } k = N \\ Lbound + (k-1)\cdot\Delta x + \Delta x/2 & , \text{ otherwise} \end{cases} \tag{12}$$

TABLE I. SEVEN BENCHMARK FUNCTIONS

	Dimension	Domain	Type	Optimum
F_{sphere}	30	[-100,100]	Min	0
F_{sumcan}	10	[-0.16,0.16]	Max	10^5
F_{twopeaks}	5	[-100,100]	Max	10.1053
$F_{\text{threepeaks}}$	5	[-100,100]	Max	10.1053
F_{chwefel}	30	[-500,500]	Min	-12569.5
$F_{\text{rastrigin}}$	30	[-600,600]	Min	0
F_{griewank}	30	[-5.12,5.12]	Min	0

5) Replacement operation

After generating a set of new individuals P' , a new population would be created by replacing some individuals of $P(t)$ with those of P'.

IV. EXPERIMENTAL STUDIES

A. Test Functions

In this section, seven benchmark functions are chosen from [8] and [13] for the test purpose, as tabulated in Table I. Among these functions, F_{sphere} is a simple unimodal function, F_{suncan} is a unimodal function with a sharp peak, $F_{twopeaks}$ and $F_{\text{threepeaks}}$ are multimodal functions with a few peaks, whereas F_{chewerf}, F_{ruitsie} and $F_{\text{rastrigin}}$ are multimodal functions with a large number of local optima.

In the experimental studies, we run all algorithms for 30 times independently, while the maximum number of fitness evaluation number for the unimodal functions is 2×10^5 and for the multimodal functions is 4×10^5. The average value and the standard deviation of the results are recorded after testing each function for the algorithm comparison. All the results of the FHEDA are generated by Visual C++ 6.0 platform using a PC with Intel 2.0 GHz CPU, 2.0 G RAM. The parameter setting of the FHEDA is listed in Table II.

B. Comparison with UMDAc, CEGDA and sur-shr-HEDA

In this part, we compare FHEDA with the UMDAc, the CEGDA and the sur-shr-HEDA. The final comparison results are summarized in Table III.

We can see from Table III that the histogram based EDAs outperform Gaussian Based EDA when solving multimodal functions with a large number of local optima. Furthermore, the performance of the FHEDA is better than that of the sur-shr-EDA for these functions. As shown in Table III, the average of the best function values of F_{schwefel}, $F_{\text{rastrigin}}$ and F_{griewank} found by the FHEDA is equal or even much better than those of the sur-shr-HEDA. For F_{sphere}, the performance of the FHEDA is also the best among these four algorithms. However, the FHEDA does not perform so good on functions with flat plateaus or deep valleys, such as F_{sumsan}, F_{twopeaks} and $F_{\text{threepeaks}}$. Parameter analysis in part *C* indicates that, with a smaller *BIN_NUM*, the algorithm can perform better for such kinds of functions.

TABLE II. PARAMETER SETTING OF THE FHEDA

Parameter	Descriptions	Value
Popsize	Size of population	400
Se	Number of excellent individuals	200
q_0	Mutation rate	0.01
BIN_NUM	Number of bins	20
ε	Rate of extending range	0.2

TABLE III. RESULT COMPARISON AMONG UMDAC, CEGDA, SUR-SHR-HEDA AND FHEDA

Function		UMDAc	CEGDA	sur-shr-HEDA	FHEDA
F_{sphere}	*avg.*	3.24×10^{-16}	3.41×10^{-6}	2.25×10^{-11}	1.11×10^{-37}
	std.	5.59×10^{-17}	8.40×10^{-7}	3.54×10^{-12}	4.23×10^{-38}
F_{sumcan}	*avg.*	221.771	99748.1	100000	4950.95
	std.	116.101	63.2197	0	11735.2
F_{twopeaks}	*avg.*	9.6327	10.0999	10.1053	9.56638
	std.	0.1073	5.92×10^{-3}	0	0.504
$F_{\text{threepeaks}}$	*avg.*	4.7331	10.1048	8.9503	5.38951
	std.	0.7406	7.99×10^{-4}	2.2063	1.260
F_{schwefel}	*avg.*	-5424.81	-5922.54	12568.6	12569.5
	std.	202.437	1893.51	0.723	1.82×10^{-12}
F_{griewank}	*avg.*	0.0581	--	1.887×10^{-16}	0
	std.	0.0335	--	5.363×10^{-17}	0
$F_{\text{rastrigin}}$	*avg.*	0.7966	--	0	0
	std.	1.1296	--	0	0

('*avg.*' is the average of the best function values found in 30 runs and '*std.* ' stands for the standard deviation)

C. Parameter Analysis

Two important parameters of the FHEDA are set as $q_0 = 0.01$ and *BIN_NUM* = 20 in previous experiments. In this part, three functions with different characteristics are used for the parameter analysis. For each parameter setting, we run the FHEDA for 30 times on these three functions respectively. The results are shown in Fig. 3 - Fig. 5, where *f*(*x*) is the average value of the results after 30 runs.

As shown in Fig. 3, the number of bins plays an important part in finding highly accurate solutions for functions like F_{sphere}. Generally, a larger *BIN_NUM* leads to better results, but the performance of the FHEDA will degrade quickly if the value of *BIN_NUM* is set too large. On the other hand, a smaller q_0 continuously leads to better results for the FHEDA. As for functions like F_{sumsan}, Fig. 4 indicates that the algorithm would perform better with a smaller *BIN_NUM.* In this case, q_0 seemingly has little effect on the results. At last, Fig. 5 shows that a larger *BIN_NUM* is preferable for multimodal functions like $F_{\text{greisenren}}$, because more bins can capture local optima more accurately when there are many local optima. The performance of the FHEDA will degrade if the value of *BIN_NUM* is set too large. Results in Fig. 5 also suggest that, q_0 should be set appropriately large to keep the diversity of the population. In conclusion, the optimum values of q_0 and *BIN_NUM* are problem dependent. How to adaptively adjust q_0 and *BIN_NUM* is a problem that needs further studies.

V. CONCLUSIONS

This paper proposed a novel Fuzzy Histogram based EDA (FHEDA) for continuous optimization. In the FHEDA, new offspring are generated from fuzzy histogram probability models. In order to gain highly accurate solutions, the estimated range of the fuzzy histogram is adjusted adaptively. Furthermore, a mutation mechanism is introduced in the sampling operation to avoid local trapping.

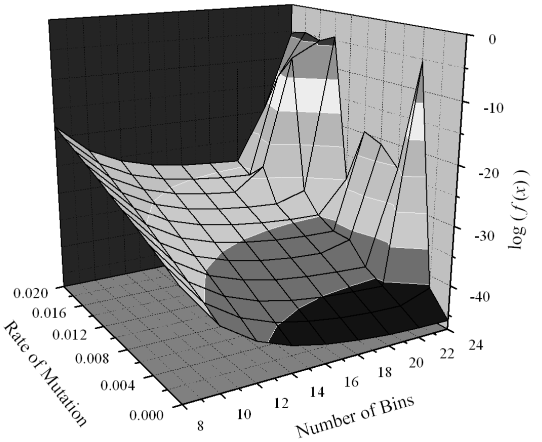

Figure 3. Influence of two parameters on the FHEDA to solve F_{sphere}.

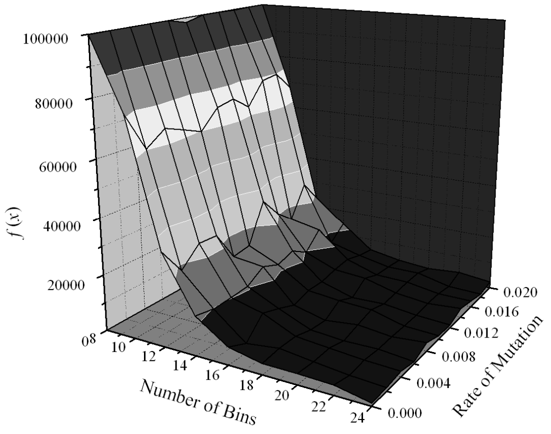

Figure 4. Influence of two parameters on the FHEDA to solve F_{sumca}

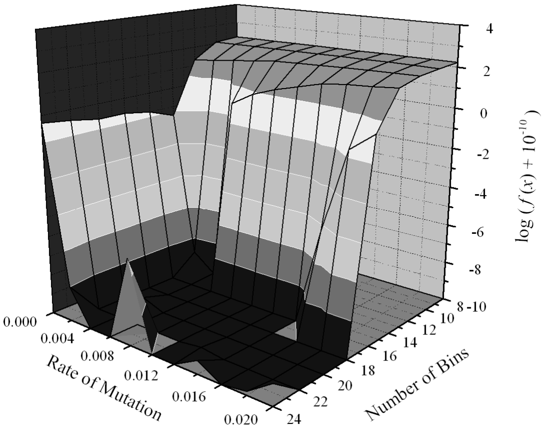

Figure 5. Influence of two parameters on the FHEDA to solve F_{griewank}

The performance of the proposed FHEDA was evaluated by testing seven benchmark functions with different characteristics. Two Gaussian based EDAs and the sur-shr-HEDA were studied for comparison. The results showed that among all experimental algorithms, the FHEDA was able to give comparatively satisfying performance on unimodal and multimodal functions. Finally, the characteristics of two important parameters in the FHEDA are also studied.

As for future work, we will further study the adaptive adjustment of parameters q_0 and *BIN_NUM* in the FHEDA. In addition, improving the FHEDA to solve continuous optimization problems with variable interdependence is another promising research area.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC) Joint Fund with Guangdong under Key Project U0835002, and the National High-Technology Research and Development Program ("863" Program) of China no. 2009AA01Z208. and in part by the Internal Competitive Research Grant, PolyU (Project Code: G-YG39).

REFERENCES

[1] S. Baluja, "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning," Technical report CMU-CS-94-163, Carnegie Mellon University, 1994.

[2] H. Mühlenbein, G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," Parallel Problem Solving from Nature - PPSN IV, Berlin, 1996, pp. 178-187.

[3] J. Zhang, W. N. Chen, J. H. Zhong, X. Tan, and Y. Li, "Continuous function optimization using hybrid ant colony approach with orthogonal design scheme," SEAL'06, LNCS 4247, 2006, pp. 126–133.

[4] I. Servet, L. T. Massuyes, and D. Stern, "Telephone network traffic overloading diagnosis and evolutionary computation techniques," in Proceedings of the Third European Conference on Artificial Evolution (AE'97), NY, G. Goos, J. Hartmanis, and J. Leeuwen (Eds.), 1997, pp. 137–144.

[5] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in Parallel Problem Solving from Nature—PPSN V, Berlin Heidelberg, 1998, pp. 418–427.

[6] P. Larrañaga, R. Etxeberria, ,J. A. Lozano, and J. M. Peňa, "Optimization in continuous domains by learning and simulation of Gaussian networks," in Proceedings of the Genetic and Evolutionary Computation Conference 2000, Las Vegas, Nevada, July 2000.

[7] P. A. N. Bosnian and D. Thierens, "Advancing continuous IDEA's with mixture distributions and factorization selection metrics," in Proc. Optimization by Building and Using Probabilistic Models (OBUPM) Workshop at the Genetic and Ecolutionary Computation Conf. (GECCO-2001), M. Pelikan and K. Sastry, Eds., San Francisco, CA, 2001, pp. 208–212.

[8] Q. Lu and X. Yao, "Clustering and learning Gaussian distribution for continuous optimization," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 35, no. 2, May 2005, pp.195–204.

[9] S. Tsutsui, ,M. Pelikan, and D. E. Goldberg, "Evolutionary algorithm using marginal histogram models in continuous domain," IlliGAL Report No. 2001019, UIUC, 2001.

[10] B. Yuan and M. Gallagher, "Playing in continuous spaces: some analysis and extension of population-based incremental learning," Proc. IEEE Congress on Evolutionary Computation, 2003, pp. 443-450.

[11] N. Ding, S. Zhou, and Z. Sun, "Optimization continuous problems using estimation of distribution algorithms based on histogram model," In Proc, the 6th Conference of Simulated Evolution and Learning, Hefei, China, 2006, PP.545-552.

[12] Q. Zhang, J. Sun, E. Tsang and J. Ford, "Hybrid Estimation of Distribution Algorithm for Global Optimsation," Engineering Computations, vol. 21, no. 1, 2004, pp. 91-107.

[13] N. Ding, S. Zhou, and Z. Sun, "Histogram-based estimation of distribution algorithm: a competent method for continuous optimization," Computer Science and Technology, vol. 23, no. 1, 2008, pp. 35-42.

[14] L. A. Zadeh, "Fuzzy sets,". Information and Control, vol. 8, 1965, pp. 338–353.

[15] J. Zhang, H. Chung, and W. L. Lo, "Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates," IEEE Transactions on Knowledge and Data Engineering, vol.20, no.7, July 2008.

[16] J. Zhang, H. Chung and W. L. LO, "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," IEEE Transactions on Evolutionary Computation, vol. 11, no. 3, June 2007, pp. 326-335.

[17] Z. H. Zhan and J. Zhang, "Adaptive particle swarm optimization". ANTS Conference 2008, LNCS 5217, pp. 227-234.

[18] K. Loquin and O. Strauss, "Fuzzy histograms and density estimation," SMPS 2006, Soft Methods in Probability and Statistics (2006), pp. 45–52.

[19] J. S. De Bonet, C. L. Isbell, and P. Viola, "MIMIC: finding optima by estimating probability densities," Advances in Neural Information Processing Systems, vol. 9 The MIT Press, Cambridge, 1997, pp. 424.

[20] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space," in Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp. 30-38.

[21] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: the Bayesian optimization algorithm," in Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, vol. 1, Orlando, FL, 1999, pp. 525-532.