

Histogram-Based Estimation of Distribution Algorithm: A Competent Method for Continuous Optimization

Nan Ding¹ (丁楠), Shu-De Zhou² (周树德), and Zeng-Qi Sun² (孙增圻)

¹*Department of Electronic Engineering, Tsinghua University, Beijing 100084, China*

²*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

E-mail: ding-n04@mails.tsinghua.edu.cn; zsd03@mails.tsinghua.edu.cn; szq-dcs@tsinghua.edu.cn

Revised November 6, 2007.

Abstract Designing efficient estimation of distribution algorithms for optimizing complex continuous problems is still a challenging task. This paper utilizes histogram probabilistic model to describe the distribution of population and to generate promising solutions. The advantage of histogram model, its intrinsic multimodality, makes it proper to describe the solution distribution of complex and multimodal continuous problems. To make histogram model more efficiently explore and exploit the search space, several strategies are brought into the algorithms: the surrounding effect reduces the population size in estimating the model with a certain number of the bins and the shrinking strategy guarantees the accuracy of optimal solutions. Furthermore, this paper shows that histogram-based EDA (Estimation of distribution algorithm) can give comparable or even much better performance than those predominant EDAs based on Gaussian models.

Keywords evolutionary algorithm, estimation of distribution algorithm, histogram probabilistic model, surrounding effect, shrinking strategy

1 Introduction

Evolutionary algorithms (EA) are of a class of stochastic black-box methods that explore the search space for optimal solutions by iteratively sampling and evaluation. Practical applications and theoretical research have shown that EA is an effective technology to optimize parameters or make decisions in the situation where there is little prior information about the objective model or the mathematic model is so complicated that conventional optimization methods often fail. Estimation of distribution algorithms (EDA) have recently become the hot topic in the field of evolutionary computation^[1–3]. The EDAs provide a novel macroscopical evolutionary paradigm, in which without any conventional operators, the population evolves by iteratively learning and sampling the probabilistic distribution model that describes the movements of population.

The core of EDAs is the probabilistic model. For 0-1 domain, since the probabilistic model built to sample the population is simple, the EDA has brought a great success. However, it is still a challenging problem for EDAs to be applied in continuous domain, even tho-

ugh several attempts have been made to extend the research results from discrete to continuous problems. Nowadays, most of the work concentrates on Gaussian probabilistic model. These include PBILc, UMDAc, EMNA, EGNA, IDEA and so on^[4–9]. Histogram-based EDAs was first put forward by Tsutsui *S et al.*^[10], which is like an extension of PBIL, splitting the searching space into a number of bins.

The predominant probabilistic model used in continuous EDAs is Gaussian probabilistic model. However, for complex optimization problems with multiple and distributed local optima, the inherent shortcoming of Gaussian-based EDA^[6,7] is that the unimodal model is too rough and thus is likely to mislead the search to a local optimum. To conquer such shortcomings of single-Gaussian model, multi-Gaussian model is adopted^[4,5,8,9]. For the multi-Gaussian model, it is usually necessary to set the number of Gaussian distribution in the model and the weights of each Gaussian model in advance. However, for practical problem with multiple local optimums, it is often hard to predict these parameters and the computational complexity increases remarkably. In comparison, histogram probabilistic model is able to represent multiple local op-

tima by different bins with different heights and thus be able to build a more correct probabilistic model compared to the Gaussian-based model in general.

Although the HEDA has those advantages as have been mentioned, however, there are still some severe drawbacks in the previous HEDA, which block the further development of the HEDA. The initial population should be large enough to sample the variables with a lot of bins, otherwise, many bins will never get a chance to be sampled; the solution accuracy is greatly determined by the width of bins and highly accurate solutions can only be achieved by setting enough number of bins.

To improve these drawbacks, several strategies are brought into our new algorithms: the surrounding effect shares the height to the surrounding bins which facilitates the algorithm to search for a wider span of good solution clusters and proves to lessen the needed size of population in estimating the model with a certain number of the bins; the shrinking strategy shrinks the search space and improves the accuracy of the optimal solution with small size of bins. Furthermore, this paper shows that histogram-based EDA with surrounding effect and shrinking strategy can give comparable or even much better performance than the predominant EDAs based on Gaussian models in the experiments.

The paper is organized as follows. In Section 2, related work of histogram based EDAs are introduced. Section 3 elaborates the surrounding and shrinking strategies in histogram based EDA. In Section 4 the experiments verify the performances of HEDA. The conclusion is given in Section 5.

2 Retrospect of HEDA

2.1 Related Work on Histogram Based EDAs

The role of probabilistic model in EDA is to describe the distribution of the promising population and to generate new populations. In continuous domain, histogram model is a straight method to graphically summarize the distribution of data. The most common form of the histogram is obtained by splitting the range of the searching space into equal-sized bins (called classes). Then for each bin, the qualities of the points belong to it decide the height of the bin. Although histograms are very simple, they are very flexible and can model complex properties like multimodality with a relatively small number of parameters. Furthermore, one does not need to assume any specific form for the underlying density function: given enough bins, a histogram estimator adapts to any kind of den-

sity as general as possible^[11].

Histogram model has already been used in previous work. Tsutsui S *et al.* proposed EDAs based on histogram model. A few data points were selected from the population and the height of each bin was simply a counter of the points belonging to it^[10,12]. Yuan Bo *et al.* improved the algorithm by utilizing all individuals' information and incorporating their fitness values^[13]. Ding Nan *et al.*'s accumulate strategy in HEDA considered both historical and current information of populations and they introduced the mutation and elitist strategy into the HEDA^[14]. All the researches are the extensions of the PBIL algorithm, which does not take the interaction relationship between variables into account. Although Bosman and Thierens^[15] developed histogram based IDEA that can model the conditional probability of each variable, it needed exponential computation as the problem dimension increases. Moreover, since the learning of the relationship was based on the biased information of an incomplete sample of the space, whether the cost of such learning was worthwhile was suspicious. Due to those reasons, this paper only studies the histogram based EDA in which no dependencies between variables are considered.

2.2 Brief Description of the Original HEDA

This work is the extension of the previous work, especially the HEDA^[14]. The general description of the HEDA is as follows.

2.1.1 Process of HEDA

- 1) Initialize the histogram model in which the bins are equal-sized and the model is normalized.
- 2) Generate population $P(t)$ using sampling method.
- 3) Evaluate and rank the population $P(t)$. Save the elitist.
- 4) Update the histogram model using accumulation learning strategy.
- 5) If the termination conditions have not been satisfied, go to step 4).

2.1.2 Accumulation Strategy

The histogram model is updated according to two kinds of information: historical and current information. For each variable i , the selected individuals will be used to construct a histogram model H_C^i . If the old histogram for variable i is denoted as H_H^i , then the height of a certain bin j of the renewed histogram

for variable i will be:

$$H^i(j) = \alpha H_H^i(j) + (1 - \alpha) \bullet H_C^i(j) \quad (1)$$

and

$$\sum_j H_C^i(j) = 1. \quad (2)$$

H_C^i is learned according to relative ranking among the qualities of the different individuals. If N best individuals of the population are selected, the k -th best individual ($k \leq N$) will make an increment of the corresponding bin which it belongs to by:

$$\Delta h_k^i = \frac{(N - k + 1)}{\sum_{l=1}^N l = \frac{2(N - k + 1)}{N(N + 1)}}. \quad (3)$$

So, $H_C^i(j) = \sum_{k=1}^N \Delta h_k^i \bullet \delta_{jk}^i$ where $\delta_{jk}^i = 1$ for $\{\delta_{jk}^i | k \in \{1, 2, \dots, N\} \wedge \min_j^i \leq v_k^i < \max_j^i\}$, otherwise $|\delta_{jk}^i| = 0$. v_k^i denotes the value of variable i of the k -th best individual, \min_j^i and \max_j^i denote the bottom and upper bounds of bin j of variable i . Updating H_C^i based on the ranking information helps improve the convergence property of the HEDA.

2.1.3 Sample with Mutation

In order to enhance the diversity of population, mutation strategy is used. In the HEDA, the mutation strategy means that each variable of an individual has a probability to be generated uniformly in the whole searching space rather than according to the probabilistic model has been learned.

2.3 Remained Drawbacks in Original HEDAs and the Motivation of New HEDAs

Experiments^[14] show the HEDA have strong ability to find the bin composing the optimum; however, there is an obvious drawback: the choice of the width of the bins is a great contradictory. If the width is set too small, it needs very large population to guarantee all the subspace can be sampled; if the width of the bin is set too large, the accuracy of the solutions would be very bad because the accuracy utterly depends on the width of histogram bin. So we have to think whether it is possible for the size of population to be reduced when estimating a model with a definite number of the bin and still have the ability to sample the whole space; and whether the accuracy of the solution can be achieved using a comparably small number of the bins. In the next section, the two strategies will be proposed to achieve our expectations, and the experiments will show that these strategies make the HEDA perform excellently.

3 sur-HEDA and shr-HEDA

3.1 Surrounding Effect in Model Learning in sur-HEDA

The sur-HEDA (the HEDA using surrounding effect) tries to expand the impact of a selected individual on the bin to which it belongs, to the bins next to its bin.

After all the $H_c(j)$ have been calculated, before accumulation, we calculate the surrounding effects on a certain bin j according to:

$$H_{\text{sur}}(j) = \beta \cdot [H_c(j - 1) + H_c(j + 1)] \quad (4)$$

where β is called the surrounding effect factor which is equal to $1 - \text{gen}/\text{MAXGEN}$ in the sur-HEDA, where gen is the current generation and MAXGEN is the maximum of generation. And the new

$$H_{c'}(j) = H_c(j) + H_{\text{sur}}(j). \quad (5)$$

After the normalization of $H_{c'}(j)$, which makes $\sum_j H_{c'}(j) = 1$, the height of bin j in the renewed histogram model is calculated by accumulation strategy:

$$H(j) = \alpha H_H(j) + (1 - \alpha) H_{c'}(j) \quad (6)$$

as mentioned in the last section.

We note that the surrounding effect can make the height of the bins be affected by its surrounding bins. This method will have a remarkable advantage to the problem where the population is small compared to the number of the bins. If the population is not large enough, after the first generation, many bins have not been sufficiently searched. In the original HEDA, those bins with few samples at first will hardly get another chance to be sampled regardless of the quality of its individuals, since the bin's height is zero. Here, we assume that in many cases, the clusters of good individuals can span several bins, using surrounding effect, only if there is one bin in this span is sampled and performs a high-quality target value, those bins beside it will also get the heights and thus have more possibility to be sampled in the next run. Gradually, those bins in the same span but with few samples will all get the heights and better chance to be sampled after next several generations. As a result, we can have more individuals sampled in that span and thus have better opportunity to find the best solution in that span.

The surrounding effect is like other heuristic method, which purposes to guide the searching through some good place that it has never been to.

Interestingly, the surrounding effect also has a potential property of hill-climbing. We build a simple 1-dimension parabola function to certify that.

Table 1. Simple Problem to Test the Hill-Climbing Ability of the sur-HEDA

Formulation	Dim	Domain	Type	Optimum
$y = x^2$	1	[0, 10]	Max	100

A 100-bin histogram model is used and the population of the sur-HEDA is set to be 2, $MAXGEN = 200$. Below is the figure of the result. The X-axis is the generation, and the Y-axis is the solution of the algorithm.

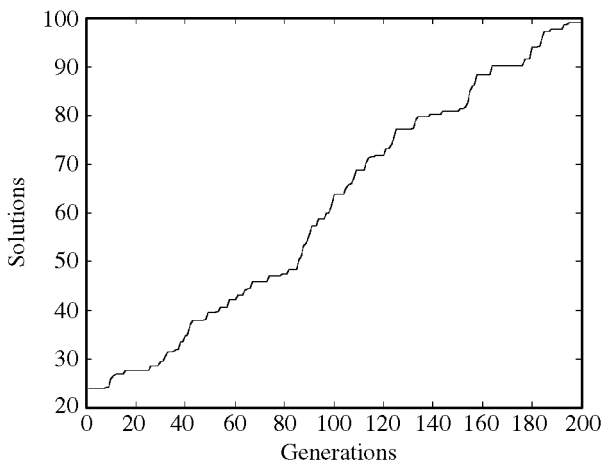


Fig.1. Hill-climbing ability of the sur-HEDA.

From Fig.1, we can find that the solution of algorithm is keeping climbing to the optimum of the problem. The force that makes it climb is from the fact that the upper bin always contains better individuals than the current bin. Because of the surrounding effect, the upper bin will have a chance to be sampled, and then a better solution from it can be found. After all the iterations, the solution gradually climbs to the optimum of the problem as a result.

3.2 Shrinking Strategy in the HEDA

Large quantities of the bins are needed for the demand of accuracy for the optimum in the original HEDA. Although the sur-HEDA is able to estimate the histogram model composed of a large quantity of the bins with a comparably small population, it is still not efficient enough for the HEDA and is still necessary to develop other cheaper way to achieve the high accuracy of the algorithm. The new way is called shrinking strategy and the HEDA using shrinking strategy is called the shr-HEDA.

In the process of the shr-HEDA, the searching span of every variable will minimize in some steps. After the new model has been built, if the highest bin of a variable is over the shrinking threshold T and also the elitist belongs to that bin, the span of the highest bin will substitute the old searching span and become the new searching span. And it will be then divided into the certain number of bins as initial step of the HEDA.

The framework of the shrinking strategy is as follows.

- 1) Find the highest bin H of variable i .
- 2) Judge if the height of H is higher than the shrinking threshold T and if the elitist belongs to it.
- 3) If the two cases are satisfied, the searching span $w(i)$ is substituted by the span of H : $w(i) \leftarrow (H)$; then, divide and initialize $w(i)$.

The shrinking strategy makes the searching process run from roughness to delicateness and be able to get an accurate value with a respectively small number of bins. The width of the bins becomes smaller and smaller during the search with the rate:

$$w(n) = w(0) \bullet \frac{1}{B^n}. \quad (7)$$

Here, $w(n)$ denotes the width of a bin after the n -th shrinking has been made, B is the number of the bins of the variable.

Furthermore, in order to guarantee the algorithm not to lose the optimum if the searching span of the algorithm is unfortunately chosen out of the position of the optimum, each sampled individual will have a mutation rate. Namely, there is a possibility of p_m for each variable of each individual to be generated randomly in the original searching space and a possibility of $1 - p_m$ for it to be generated according to the histogram model. Clearly, it is just an extension of the mutation mechanism in the original HEDA.

The situation becomes somewhat more complicated, if an individual generated by mutation, that is, at least one variable of the individual is out of the current span of the histogram model, is good and chosen. In this case, these variables which are out of the space of the models have to be reset to the original histogram model (spanning the whole space). The heights of the bins of the reset model will be generated according to the selected individuals. Note that although this mutation step might reduce the efficiency of the shrinking strategy in this case, however, it helps avoid being trapped in local optima so as to improve the global solution of the algorithm. Thus, the global performance of the algorithm can be improved as a whole.

3.3 Combination of Surrounding Effect and Shrinking Strategy with HEDA

The surrounding effect reduces the needed population in a model and the shrinking strategy changes the span of the model from wide to narrow. The two improved methods are not mutually incompatible. Thus, as the experiments in the next section will show that, the combination of the surrounding effect and shrinking method in HEDA will bring some wonderful performances on many continuous problems. In the following experiment, we will call the HEDA without using surrounding effect and shrinking method, the sHEDA; the HEDA simply using surrounding effect, the sur-HEDA; the HEDA simply using shrinking method, the shr-HEDA; and the HEDA using both the surrounding effect and the shrinking method, the sur-shr-HEDA.

There is another thing we have to mention here. Since the surrounding effect will make the bins next to the selected bins grow, after normalization, the height of the highest bin will reduce greatly compared to the height if surrounding effect is not taken account. Thus, if the surrounding effect is too big in the whole process of the algorithm, it will make the sur-shr-HEDA unable to shrink, since the threshold value is hardly to reach for any bin. Thus, the surrounding effect factor should be set to reduce so as to avoid the barrier of the shrinking strategy.

In the sur-shr-HEDA, the surrounding effect is related to the age of the searching span of the model of each variable i .

$$\begin{aligned} \beta(i) &= \beta(\text{Mod}N(i)) \\ &= \begin{cases} 1 - \text{Mod}N(i)/K, & 1 \leq \text{Mod}N(i) < K, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (8)$$

in which K is the factor of the limited age to have surrounding effect and β is the strength of the impact. The age denotes the number of generations that the searching span has experienced.

4 Experimental Results and Analysis

Two groups of experiments are carried out: one is to verify the ability of the HEDA in finding the right bin which contains the global optimal solution; the other is to compare the performance of our histogram-based algorithms with several advanced Gaussian-based algorithms in solving some typical continuous problems.

4.1 Ability of Searching Optimal Area

As mentioned, the shrinking strategy is powerful in getting the precise optimum if the algorithm can correctly find the bins. So, it is important for the HEDA

to find the bin that composes of the optimum. This experiment is to compare the ability of the sHEDA and the sur-HEDA in finding the area where the optima locate.

4.1.1 Experimental Settings

Several well-known continuous test functions, including the Schwefel function, the Griewank function, and the two-peak function, are used to test the performance of sHEDA and sur-HEDA. Table 2 lists the parameters of the three test functions and their respective optima. The description of these problems can refer to [10, 14].

Table 2. Test Functions

	Dim	Domain	Type	Optimum
Schwefel ⁽¹⁾	5	[-2, 2]	Min	0
Griewank	10	[-5, 5]	Min	0
Two-Peak ⁽¹⁾	10	[0, 12]	Min	0

Note: The superscript on Schwefel⁽¹⁾ and Two-peak⁽¹⁾ is used to distinguish from those homonymous functions in Subsection 4.2.

The width of the bin is set to 0.1, and the optimal area refers to the bin which the optimal solution belongs to. We say it is successful detection of the optimal area if one or more individuals are sampled in the optimal area within the given number of function evaluations. The population size is set to 10, 50, 100 and 200 for each problem and 20 independent runs are performed for each parameter setting. Each run continues until the global optimal area is found or a maximum of 50,000 function evaluations is reached. In both algorithms, the initial population is generated uniformly. The parameter settings of sHEDA and sur-HEDA are the same: the width of bin is set to 0.1, mutation rate is set to 0.05, accumulation rate 0.2, 50% of population is selected for model updating and elitist strategy is used.

4.1.2 Experimental Results

Table 3 shows the experimental results of the comparison of the sHEDA and the sur-HEDA. #OPT is the number of runs in which algorithm succeeds in finding the global optimal area and MNE denotes the mean number of function evaluations (MNE) to find the global optimal area in those successful runs.

4.1.3 Experimental Analysis

Our purpose here is to observe the difference between the relationships of population and the abilities of achieving a given accuracy for sHEDA and sur-HEDA. After an overlook of the result of the experi-

Table 3. Comparison Between HEDA and sur-HEDA in Finding Optimal Area

		Population Size							
		10		50		100		200	
		OPT#	MNE	OPT#	MNE	OPT#	MNE	OPT#	MNE
sHEDA	Schwefel ⁽¹⁾	10	21 056.3	20	4 741.3	20	2 607.5	20	3 221.4
	Griewank	19	8 625.3	20	9 005.7	20	6 164.3	20	5 709.4
	Two-Peak ⁽¹⁾	8	16 596.5	20	6 140.0	20	6 330.1	20	4 213.1
sur-HEDA	Schwefel ⁽¹⁾	20	421.2	20	651.4	20	1 193.1	20	1 914.3
	Griewank	20	1 199.0	20	2 574.2	20	4 251.2	20	5 212.7
	Two-Peak ⁽¹⁾	20	26 500.3	20	4 180.7	20	5 432.2	20	8 556.1

ment, it is clear that sur-HEDA always performs better than sHEDA when the population is comparably small. Furthermore, the numbers in bold font in Table 3 mean the fewest average function evaluations for the certain algorithm needed to get a full 20-success in solving a certain problem. The sur-HEDA performs best when the population is 10 or 50; the sHEDA performs better when the population is 100 or 200.

When the population gets larger, the stabilities and the speed of the sHEDA to find the optimal area is remarkably increasing. No full success of the three problems is made when the population is 10, but all the other 3 bigger sizes of population get the full success. The reason that the sHEDA sometimes fails in a small population is because its inability to estimate the bins where there has been no individual sampled, and this experimental result is conformed to our discussion in Section 2 on the drawback of sHEDA. Thus, it is concluded that sHEDA requires a comparably big population to ensure its stability in finding the right optimal area.

On the other hand, the sur-HEDA performs greatly when the population is 10 on both Schwefel and Griewank problems, whereas, performs worse with population 50 on the Two-peak problem. However, as a whole, we still find the sur-HEDA is much more powerful than the sHEDA in searching the optimal area stably with a small number of individuals compared to the number of the bins. It is because the sur-HEDA has the ability to estimate the bins surrounding the bin of the selected individuals, which remarkably improve the learning efficiency of the sHEDA when the population is small.

Since the sur-HEDA performs nicely with comparably small population, in the next experiment of searching for the optima, we can use only 1/10 number of the individuals as the sHEDA for each generation, which means the number of the generations of the sur-HEDA can be 10 times of that of the sHEDA when the number of the function evaluation is fixed. In fact, the difference of the population and generation makes some exciting results of the sur-HEDA in the next experi-

ment. Furthermore, the second strategy — shrinking strategy will be introduced in both sHEDA and sur-HEDA in the next experiment.

4.2 Ability of Searching Optimum

4.2.1 Experimental Settings

Two HEDA approaches, the shr-HEDA and the sur-shr-HEDA, and two Gaussian EDA approaches, UMDA and CEGDA are used on the five test functions chosen from [16] for comparison purposes. Table 4 lists the settings for those five functions. We choose these five functions because they have different characteristics on the number of optima. In those functions, the Sumcan function, Two-peak function, Three-peak function are non-separable. The Sphere function and the Schwefel function are separable.

Table 4. Test Functions

	Dim	Domain	Type	Optimum
Sphere	30	[-100, 100]	Min	0
Sumcan	10	[-0.16, 0.16]	Max	10 ⁵
Two-peak ⁽²⁾	5	[-100, 100]	Max	10.1053
Three-peak	5	[-100, 100]	Max	10.1053
Schwefel ⁽²⁾	30	[-500, 500]	Min	-12569.5

Note: The superscript on Schwefel⁽²⁾ and Two-peak⁽²⁾ is used to distinguish from those homonymous functions in Subsection 4.1.

For each algorithm, the parameter settings on the five functions are unchanged. All the results are made after 30 independent runs and the maximal evaluation number for the unimodal functions is 2×10^5 and for the multimodal functions 4×10^5 respectively. The initial population is generated uniformly at random in the specified domain of each function. For the shr-HEDA and the sur-shr-HEDA, mutation rate is set to 0.05, accumulation rate is 0.2, the number of the bins is set to 99, the threshold value is set to 0.9, and elitist strategy is used. The population size in the shr-HEDA is set to 1000, while the population size in the sur-shr-HEDA is 100. For the purpose of fair comparison,

the results obtained by Gaussian-based algorithms reported in [16] on these problems are directly used, and their detailed experimental settings can refer to [16].

4.2.2 Experimental Results

All of the results are summarized in Tables 5–9.

Table 5. Experimental Results for the Sphere Function

	Best	Mean	Std.
UMDA _c	1.88×10^{-16}	3.24×10^{-016}	5.59×10^{-17}
CEGDA	3.38×10^{-8}	3.41×10^{-6}	8.40×10^{-7}
shr-HEDA	4.3624×10^{-2}	3.147	2.815
sur-shr-HEDA	1.87×10^{-11}	2.25×10^{-11}	3.54×10^{-12}

Table 6. Experimental Results for the Sumcan Function

	Best	Mean	Std.
UMDA _c	698.72	221.771	116.101
CEGDA	99 834.5	99 748.1	63.2 197
shr-HEDA	12 912.1	1 426.9	4 136.6
sur-shr-HEDA	100 000	100 000	0

Table 7. Experimental Results for the Two-Peak⁽²⁾ Function

	Best	Mean	Std.
UMDA _c	10.105 3	9.632 7	0.107 3
CEGDA	10.105 3	10.099 9	5.92×10^{-3}
shr-HEDA	10.105 3	6.352 6	3.458 5
sur-shr-HEDA	10.105 3	10.105 3	0

Table 8. Experimental Results for the Three-Peak Function

	Best	Mean	Std.
UMDA _c	5.187 7	4.733 1	0.740 6
CEGDA	10.105 3	10.104 8	7.99×10^{-4}
shr-HEDA	6.684 8	4.246 2	1.362 2
sur-shr-HEDA	10.105 3	8.950 3	2.206 3

Table 9. Experimental Results for the Schwefel⁽²⁾ Function

	Best	Mean	Std.
UMDA _c	-5 928.24	-5 424.81	202.437
CEGDA	-8 712.31	-5 922.54	1 893.51
shr-HEDA	-12 569.5	-12 560.2	10.512
sur-shr-HEDA	-12 569.5	-12 568.6	0.723

4.2.3 Experimental Analysis

These five test functions can be divided into three groups: the Sphere function and the Sumcan function are unimodal functions, the Two-peak and the Three-peak functions are multimodal functions with a few optima, and the Schwefel is multimodal function with many optima.

1. Unimodal Functions

The results of the Sphere function and the Sumcan function are listed in Tables 5 and 6. Basically, these two functions are easy to solve, especially for the Sphere function, three of the four algorithms performs quite well. The UMDA performs badly in the Sumcan function. Lu *et al.*^[16] analyzes the reason for that and believes that the Sumcan function has strong variable interdependencies which is ignored by the UMDA.

Furthermore, the shr-HEDA performs really badly in solving the Sphere and the Sumcan problems, whereas the sur-shr-HEDA performs quite well. The reason is that the sur-shr-HEDA has the ability of hill-climbing and all individuals of different locations can do the local hill-climbing search simultaneously. It helps the sur-shr-HEDA to immediately capture the bin which the optimum belongs to and gets ready for the shrinking strategy. For the shr-HEDA, however, the learning of the height of the bins is simply based on sampled individuals. It means if the optimal area of one variable is only with a very small number of individuals, it is likely to be overwhelmed by other bins since the other variables of the individuals might not be good enough to combine together a strong individual as a whole.

2. Multimodal Functions with a Few Optima

Tables 7 and 8 demonstrates the results of the Two-peak function and the Three-peak function. These two functions have deceptive local optima to mislead the algorithm. In all the four algorithms, the CEGDA performs best and the sur-shr-HEDA can also find the optimum in some runs. The UMDA cannot find the optima from time to time because there is only one Gaussian distribution made for each variable which can be easily misled by the deceptive peak. In comparison, the reason that the CEGDA outperforms the UMDA is clear: the CEGDA has the ability to cluster firstly the individuals to recognize the small number of the optima and then search for each optimum respectively just as doing two or three unimodal problems.

In this time, we again notice the different performances of the sur-shr-HEDA and the shr-HEDA. In fact, the reason is still the ability of hill-climbing for the sur-shr-HEDA. For example, for the Two-peak function, two groups of the individuals are separated in

the beginning of the algorithm to the belonging of the two peaks. Then each group makes the hill-climbing to their own peaks and then the highest will win the competition in the end. For the Three-peak function, we notice that all the failure results are all because the sur-shr-HEDA is misled into a wrong peak. It is probably due to the distribution of initial individuals are sampled not ubiquitous and equal in the searching space which makes only a few individuals can hill-climb on the correct peak. An increase of the population or the equal choice of the initial individuals in searching space might improve the performance on the Three-peak function for the sur-shr-HEDA.

3. Multimodal Functions with Many Optima

The results of the Schwefel function are listed in Table 9. Both the UMDA and the CEGDA perform badly, whereas the shr-HEDA and the sur-shr-HEDA perform quite well. The reason for the badness of the UMDA is still that its only one Gauss distribution is far from enough to find the general optimum in the wide area of the local optima. The CEGDA performs badly because of its inability to make so many clusters, which would otherwise take an expensive computational cost. As a result, in [16], Lu admitted that the Gaussian-based EDA requires an impractical population size to ensure finding the optimum. The HEDAs, on the other hand, performs wonderfully on such problems, since the number of the bins can match the optima of the searching space and the cost for this complexity is much smaller according to our description of the algorithm. We then have some more experiments to certify the wonderful performances of the HEDAs in solving the multimodal functions with many optima.

4.2.4 More Experiments on Multimodal Functions with Many Optima Using the HEDAs

Now let us focus on the 3rd class problems and do some more experiments between shr-HEDA and sur-shr-HEDA. Two test problems, the Griewank function and the Rastrigin function from [14], are used and their descriptions in this experiment are listed in Table 10. Of the two functions, the Griewank function is non-separable.

Table 10. Test Functions with Many Optima

	Dim	Domain	Type	Optimum
Griewank	30	[-600, 600]	Min	0
Rastrigin	30	[-5.12, 5.12]	Min	0

The parameter settings of the two algorithms are the same: mutation rate is 0.05, accumulation rate is 0.2. The number of the bins is 99. The population

is 1000. The maximum generation is 400. 20% of the population is selected for model updating. The threshold value is set to 0.9. Elitist strategy is used. For the sur-shr-HEDA, $K = 800$.

The experimental results are shown in Tables 11 and 12.

Table 11. Experimental Results for the Griewank Function

	Best	Mean	Std.
shr-HEDA	0.0276	0.0581	0.0335
sur-shr-HEDA	1.110×10^{-16}	1.887×10^{-16}	5.363×10^{-17}

Table 12. Experimental Results for the Rastrigin Function

	Best	Mean	Std.
shr-HEDA	2.152×10^{-4}	0.7966	1.1296
sur-shr-HEDA	0	0	0

Note that we set the population of both the two algorithms to be 1000 and then the only difference between the two algorithms is whether or not the surrounding effect is accounted. In solving these two functions, the sur-shr-HEDA outperforms the shr-HEDA. In addition, the sur-shr-HEDA perfectly solves these two multimodal problems in high accuracy. It again shows us that the surrounding effect is an efficient method to improve the HEDA and the sur-shr-HEDA is a wonderful tool to solve the multimodal continuous problems.

4.2.5 More Comments on Non-Separable Problem Using the HEDAs

Among the non-separable functions we use in this algorithm, except for the Three-Peak function, the sur-shr-HEDA has got great performances on them which even outperforms the CEGNA that concerns linkage information. But the instable performance in Three-Peak function is not mainly due to its losing information on variable linkage in our experiment as has been mentioned. Thus, although the sur-shr-HEDA does not contain linkage information, however, since a thorough searching is performed on each variable of the problem, in some cases, the algorithm is able to handle non-separable problems.

5 Conclusion

In this paper, we aim to discuss and improve the continuous estimation of distribution algorithms based on histogram probabilistic model. The advantages of histogram model in optimizing complex and multimodal functions are analyzed. To make histogram

model efficiently explore and exploit the search space, two strategies are brought into the algorithms and is certified to work well by the experiments. The surrounding effect reduces the population size in estimating the model with a certain number of the bins and the shrinking strategy guarantees the accuracy of optimal solutions. The proposed sur-shr-EDA can optimize several complex multimodal functions very successfully.

This study has great implication for designing continuous EDAs. In the EDAs literature, Gaussian probabilistic model occupies a predominant position, and most of the research concentrates on how to adapt the general Gaussian model to fit complex continuous landscapes. This paper takes a successful attempt to use other probabilistic model. The experiment shows that histogram-based EDA gives comparable and even much better performance than those predominant EDA methods based on Gaussian related models. The experimental comparison with the most recently published results of Gaussian-based EDAs indicates that histogram model is potential alternative for designing competent continuous EDAs.

More research is needed to find applications of the histogram probabilistic model in developing efficient continuous optimization algorithms.

References

- [1] Larrañaga P, Lozano J A. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, 2002.
- [2] Pelikan M. Hierarchical Bayesian Optimization Algorithm: Toward A New Generation of Evolutionary Algorithms. Springer-Verlag, 2005.
- [3] Očenášek J. Parallel estimation of distribution algorithms [Dissertation]. Brno University of Technology, 2002.
- [4] Larrañaga P, Etxeberria R, Lozano J A, Peña J M. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, University of the Basque Country, 1999.
- [5] Larrañaga P, Etxeberria R, Lozano J A, Peña J M. Optimization in continuous domains by learning and simulation of Gaussian networks. In *Proc. the Genetic and Evolutionary Computation Conference*, Las Vegas, Nevada, 2000, pp.201–204.
- [6] Sebag M, Ducolombier A. Extending Population-Based Incremental Learning to Continuous Search Spaces. *Parallel Problem Solving From Nature — PPSN V*, Springer-Verlag, 1998, pp.418–427.
- [7] Larrañaga P, Lozano J A, Bengoetxea E. Estimation of Distribution Algorithms based on multivariate normal and Gaussian networks. Technical report KZZA-IK-1-01, University of the Basque Country, 2001.
- [8] Bosman P, Thierens D. Expanding from Discrete to Continuous Estimation of Distribution Algorithms: IDEA. *Parallel Problem Solving From Nature — PPSN VI*, 2000, pp.767–776.
- [9] Ahn C W. Real-coded Bayesian optimization algorithm. In *Proc. Advances in Evolutionary Algorithms: Theory, Design and Practice*, 2006, pp.85–124.
- [10] Tsutsui S, Pelikan M, Goldberg D E. Evolutionary algorithm using marginal histogram models in continuous domain. In *Proc. the 2001 Genetic and Evolutionary Computation Conference Workshop*, San Francisco, CA, 2001, pp.230–233.
- [11] Petri K, Petri M. Information-theoretically optimal histogram density estimation. HIIT Technical Reports 2006-2, Helsinki Institute for Information Technology, 2006.
- [12] Pelikan M, David E G, Tsutsui S. Combining the strengths of the Bayesian optimization algorithm and adaptive evolution strategies. IlliGAL Report No. 2001023, 2001.
- [13] Yuan B, Gallagher M. Playing in continuous spaces: Some analysis and extension of population-based incremental learning. In *Proc. IEEE Congress on Evolutionary Computation*, 2003, pp.443–450.
- [14] Ding N, Zhou S, Sun Z. Optimizing 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.
- [15] Bosman P, Thierens D. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Proc. the Optimization by Building and Using Probabilistic Models OBUPM Workshop at GECCO-2000*, San Francisco, California, 2000, pp.197–200.
- [16] Lu Q, Yao X. Clustering and learning Gaussian distribution for continuous optimization. *IEEE Transactions on Systems, Man and Cybernetics-Part C*, 2005, 35(2): 195–204.



Nan Ding is an undergraduate student of the Department of Electronic Engineering at Tsinghua University, Beijing, China. His research interest mainly focuses on machine learning, probabilistic graphical models and evolutionary computation.



Shu-De Zhou is a Ph.D. candidate in State Key Lab of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University. He received his B.S. degree from Xidian University in 2003. His research interest mainly focuses on evolutionary computation.



Zeng-Qi Sun is a professor in State Key Lab of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University. His research interests include robotics, intelligent control, evolutionary computation, fuzzy system and neural networks.