Clustering and Learning Gaussian Distribution for Continuous Optimization

Qiang Lu and Xin Yao, *Fellow, IEEE*

***Abstract*—Since the Estimation of Distribution Algorithm (EDA) was introduced, different approaches in continuous domains have been developed. Initially, the single Gaussian distribution was broadly used when building the probabilistic models, which would normally mislead the search when dealing with multimodal functions. Some researchers later constructed EDAs that take advantage of mixture probability distributions by using clustering techniques. But their algorithms all need prior knowledge before applying clustering, which is unreasonable in real life. In this paper, two new EDAs for continuous optimization are proposed, both of which incorporate clustering techniques into estimation process to break the single Gaussian distribution assumption. The new algorithms, Clustering and Estimation of Gaussian Network Algorithm based on BGe metric and Clustering and Estimation of Gaussian Distribution Algorithm, not only show great advantage in optimizing multimodal functions with a few local optima, but also overcome the restriction of demanding prior knowledge before clustering by using a very reliable clustering technique, Rival Penalized Competitive Learning. This is the first time that EDAs have the ability to detect the number of global optima automatically. A set of experiments have been implemented to evaluate the performance of new algorithms. Besides the improvement over some multimodal functions, according to the No Free Lunch theory, their weak side is also showed.**

***Index Terms*—Clustering, estimation of distribution algorithm, Gaussian distribution.**

I. Introduction

ESTIMATION OF Distribution Algorithms (EDAs) are population based search algorithms that generate new population from the estimated distribution based on current promising solutions. Compared with the other evolutionary algorithms (e.g., GA), the crucial change is that both crossover and mutation operators are substituted by the estimation process. The effectiveness of EDAs have been evaluated by many works, both in combinatorial optimization [1]–[8] and in continuous optimization [9]–[15]. Basically, continuous optimization by EDAs can be summarized into the following framework.

1) Initialize a population of N individuals randomly $\rightarrow D_0$.
2) Select $\mathrm{Se} \leq N$ individuals from $D_{l-1} (l = 1, 2, \ldots)$ according to a selection method $\rightarrow D_{l-1}^{\mathrm{Se}}$.
3) Estimate n-dimensional probability density function (pdf) based on $D_{l-1}^{\mathrm{Se}} \rightarrow p_l(\vec{x}) = p(\vec{x} \mid D_{l-1}^{\mathrm{Se}})$.
4) Sample N new individuals from $p_l(\vec{x})$, form new population by partially or fully replacing the current population $\rightarrow D_l$.
5) Stop if some stopping criterion is reached, go to step 2 otherwise.

Early EDAs for continuous optimization presumed that selected vector is a random sample from a single Gaussian distribution. This assumption makes these algorithms unsuccessful when dealing with multimodal problems. For a multimodal function, the promising solutions selected by some fitness based selection scheme normally tend to form several groups, and each group encompasses one of the local optima. Under this condition, the estimated distribution—according to this single Gaussian distribution assumption—will be far from the real one. Thus, new populations generated based on this wrong estimation would be likely to be in the wrong search area, and either resulting in a slow convergence or more seriously misleading the search to a local optimum other than the global optimum. Some researchers have noticed this problem and constructed several EDAs using Gaussian mixture distributions [16]–[19] by incorporating clustering techniques.

Two new EDAs, taking advantage of data clustering techniques to break the single Gaussian distribution assumption, are proposed in this paper. Unlike previous EDAs that also use mixture distributions, the clustering approaches used in the proposed EDAs do not need prior knowledge to work. Both algorithms are evaluated on a set of selected functions for performance assessment. The results show that they can perform very well when dealing with multimodal functions that do not contain too many local optima. And on the other hand, due to the use of a Rival Penalized Competitive Learning (RPCL) clustering technique [20], [21], for the first time EDAs have the ability to detect the number of global optima automatically. According to the No Free Lunch theory [22], the weakness of the new algorithms are also analyzed. The new algorithms are denoted as "Clustering and Estimation of Gaussian Network Algorithm based on BGe metric" ($\mathrm{CEGNA_{BGe}}$) and "Clustering and Estimation of Gaussian Distribution Algorithm" (CEGDA) respectively in this paper.

The rest of this paper is organized as follows. Section II gives a brief review of EDAs for continuous optimization, and illustrates why clustering is needed and how it can be used to break the single Gaussian distribution assumption. Section III introduces RPCL and shows its automatic number selection ability by comparing with another clustering approach, k-Means. Section IV presents two new EDAs: $\mathrm{CEGNA_{BGe}}$ and CEGDA. Section V presents the experimental results on a set of functions to assess the performance of the proposed EDAs. Section VI gives the final conclusion and discusses the topics for future research.

Manuscript received August 31, 2003; revised February 29, 2004 and April 21, 2004. This paper was recommended by Guest Editor Y. Jin

The authors are with the School of Computer Science,University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: Q.Lu@cs.bham.ac.uk; X.Yao@cs.bham.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2004.841914

II. Breaking Single Gaussian Distribution Assumption by Clustering

The EDA was first introduced as an approach for combinatorial optimization with binary string representation. While not long after this new pattern was created, various EDAs for continuous optimization were constructed.

Based on the complexity of the probabilistic model used for learning the distribution of the selected individuals, EDAs can be categorized into four classes. Algorithms belonging to the first class assume that the variables are independent of each other and the joint probability density function for n-dimensional vector $\vec{x} = (x_1, \ldots, x_n)$ can be factorized according to $p(\vec{x}) = \prod_{i=1}^{n} p(x_i)$, e.g., $\mathrm{PBIL_c}$ [9], SHCLVND [10], $\mathrm{UMDA_c}$ [13]; in class two, pairwise dependencies are considered, which means given a permutation $\pi = (i_1, \ldots, i_n)$, the joint density function can be represented as the product of one univariate density function and $n-1$ pairwise conditional density functions, denoted as $p(x) = p(x_{i_1}) \cdot p(x_{i_2} \mid x_{i_1}) \cdot \cdots \cdot p(x_{i_n} \mid x_{i_{n-1}})$, such as $\mathrm{MIMIC_c}$ [13]; approaches of the third class take any multivariate interdependencies into account, the examples are $\mathrm{EGNA_{BGe}}$, $\mathrm{EGNA_{ee}}$ [12], [13] *et al.* EDAs of the last class use mixture distributions for conquering multimodal functions, like IDEA [19].

EDAs in the first three classes introduced above have a common feature, which is they all assume that the data used for estimation comes from one single Gaussian distribution. This simplification, on one hand, makes it easy to import some theories for building the model, but on the other hand, the drawback is obvious. If the distribution of those selected individuals is far from a single Gaussian distribution, the result of estimation will definitely mislead the search along a wrong direction.

The shape of a Gaussian distribution is like that of a sphere with the density increasing from the periphery to the core. Because multimodal functions have multiple local optima, intuitively the selected better fitness individuals will not just surround one position, but disperse to form several separated groups. So from this point of view, multimodal functions are the most probable unsuitable cases for single Gaussian distribution assumption.

A simple constructed bimodal function is used to illustrate this deduction. Considering minimizing the following function:

$$F_{\mathrm{Test}}(\vec{x}) = \begin{cases} (x_1-5)^2 + (x_2+5)^2 & x_1 \geq 0 \\ (x_1+5)^2 + (x_2-5)^2 & x_1 < 0 \end{cases} \quad (1)$$

where $\vec{x} = (x_1, x_2)$. Apparently it has two local optima, $(5, -5)$ and $(-5, 5)$, which are also the global minima. Selecting 1000 individuals from a randomly generated population in $[-10, 10]$ (population size $= 5000$)[1], and the distribution is shown in Fig. 1. There are two blocks of data, each of which surrounds one of the global minima. First, the estimation method in $\mathrm{UMDA_c}$ [13] was used. It assumes that the distribution is a single Gaussian and the variables are independent of each other. Based on its estimation result, which is $\mathcal{N}((-0.097, -0.063), \left[\begin{smallmatrix} 5.28 & 0 \\ 0 & 5.36 \end{smallmatrix}\right])$, 1000 new individuals are generated by simulation, and the distribution is shown in Fig. 2. Obviously, the estimation is far from the real distribution of the data set.

[1]The purpose of using this large population size is only for the clarity of illustration.

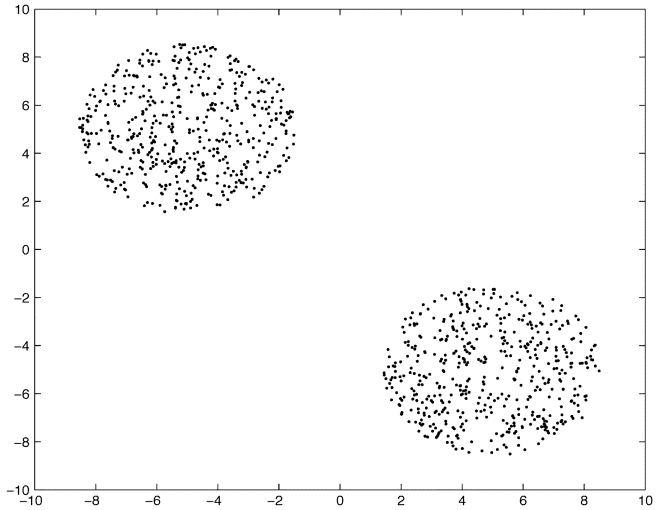

Fig. 1. Distribution of selected 1000 individuals based on the bimodal function (1).

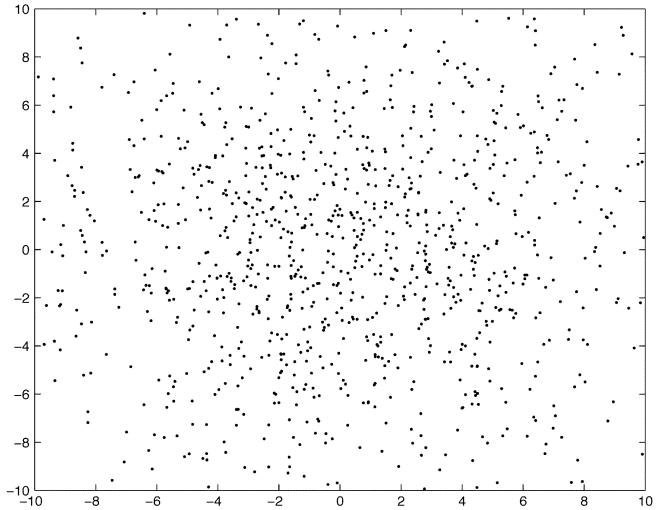

Fig. 2. Distribution of generated 1000 individuals based on the estimation over data in Fig. 1 using $\mathrm{UMDA_c}$.

From this simple case, it can be found that when optimizing multimodal problems, if the single Gaussian distribution assumption applies, the estimated distribution can be very poor. In order to overcome this deficiency, clustering techniques are incorporated.

Clustering was used with evolutionary algorithms and some particular EDA algorithms to alleviate the difficulty derived from the existence of symmetry structure in the work of Pelikan and Goldberg [17]. By clustering the selected individuals and processing each cluster separately, the presented approach successfully solves a list of combinatorial problems, e.g., two-max, graph bisection *et al.* Some other researchers also used clustering in their EDAs for continuous optimization to build mixture distributions.

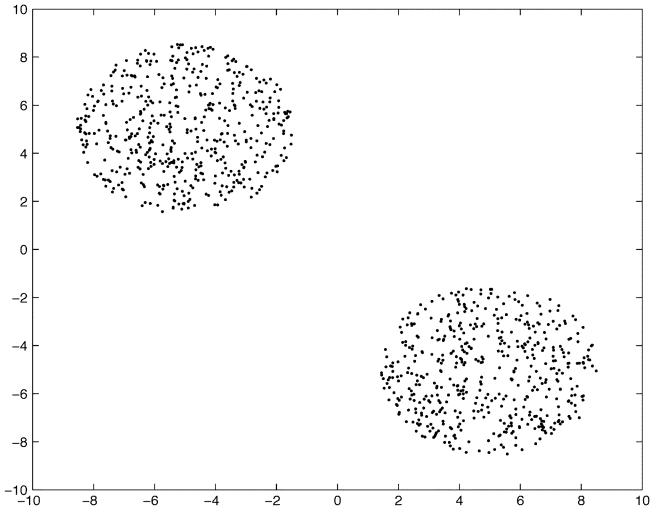

Fig. 3. Distribution of selected 1000 individuals based on the bimodal function (1).

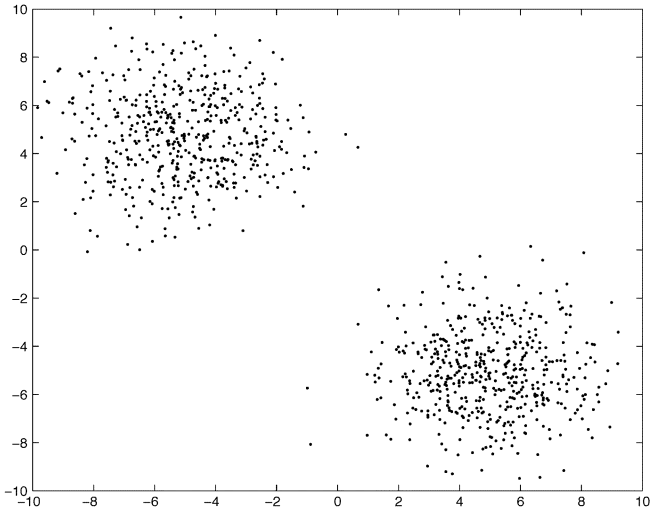

Fig. 4. Distribution of generated 1000 individuals based on the clustered estimation over data in Fig. 3 using $\mathrm{UMDA_c}$.

So, if we regard the individuals in Fig. 3 as two clusters, the estimated distribution for cluster D_1 is $\mathcal{N}((-5.04, 4.99), \left[\begin{smallmatrix}1.81 & 0\\ 0 & 1.78\end{smallmatrix}\right])$, and the distribution for cluster D_2 is $\mathcal{N}((4.87, -5.13), \left[\begin{smallmatrix}1.83 & 0\\ 0 & 1.75\end{smallmatrix}\right])$. Then another 1000 individuals are generated and the distribution is illustrated in Fig. 4. The accuracy of the estimated distribution is greatly enhanced.

By using clustering the joint probability density function of the selected solutions is assumed to be a finite mixture of multivariate Gaussian density functions. That is

$$p(\vec{x}) = \sum_{i=1}^{k} \alpha_i p_i(\vec{x}), \quad \sum_{i=1}^{k} \alpha_i = 1, \quad \alpha_i \geq 0 \tag{2}$$

where $p_i(\vec{x})$ is a single multivariate Gaussian joint density function.

III. Prior Knowledge for Clustering

In previous works, clustering approaches normally required prior knowledge, either a predefined cluster number or an estimated minimal distance between different clusters. In [17], [19], a well-known approach, k-Means, was used as the clustering tool. But the number of clusters must be provided before the clustering procedure can be executed. For a practical function, knowing such prior knowledge is usually impossible and unfeasible.

A different clustering technique, RPCL, is used in the proposed algorithms. It can detect the real number of clusters during the clustering process without any prior knowledge. In this following, a comparison between k-Means and RPCL is made. Before showing the comparison results, a quick look at the implementation of these two clustering approaches is necessary.

A. K-Means Clustering

In k-Means clustering, each cluster is specified by its center vector. For any given number k, the clustering process can be summarized as follows.

1) Generate k centers $\vec{m}_1, \ldots, \vec{m}_k$ randomly.
2) Assign each individual to the nearest center.
3) Update each center to the mean vector of all the individuals assigned to it in last step.
4) If the new center is different to the previous one, repeat from step 2.
5) Otherwise, stop clustering and return the cluster centers.

B. RPCL

RPCL is a heuristic competitive learning algorithm proposed for clustering, with the favorable feature that it can select the number of clusters during learning automatically [20]. The key idea of RPCL is that for each input $\vec{x}$, not only the winner is updated by a learning rate α_c to approach $\vec{x}$, but also the second winner (rival) is de-learned by a de-learning rate α_r. That is

$$\begin{aligned}
\vec{m}_c^{(t+1)} &= \vec{m}_c^{(t)} + \alpha_c(\vec{x} - \vec{m}_c),\\
c &= \arg\min_j \gamma_j \|\vec{x} - \vec{m}_j\|^2\\
\vec{m}_r^{(t+1)} &= \vec{m}_r^{(t)} - \alpha_r(\vec{x} - \vec{m}_r),\\
r &= \arg\min_{j \neq c} \gamma_j \|\vec{x} - \vec{m}_j\|^2\\
\vec{m}_j^{(t+1)} &= \vec{m}_j^{(t)}, \quad j \neq c, \quad j \neq r
\end{aligned} \tag{3}$$

where $\vec{m}_j$ represents the center of the cluster $j, \alpha_c > \alpha_r$, and γ_j being the frequency that $\vec{m}_j$ wins the competition up to now. The mechanism of RPCL tries to push its rival far away from the cluster toward which the winner is moving, thus implicitly produces a force which attempts to make sure that each cluster is learned by only one vector. Because the rival may belong to other clusters, de-learning rate usually is much smaller than learning rate. The effect of using this mechanism is that the appropriate number of units will be selected automatically to represent an input data set by gradually driving extra units far away from the data set when the number of units used for learning is larger than the number of clusters in the input data set.

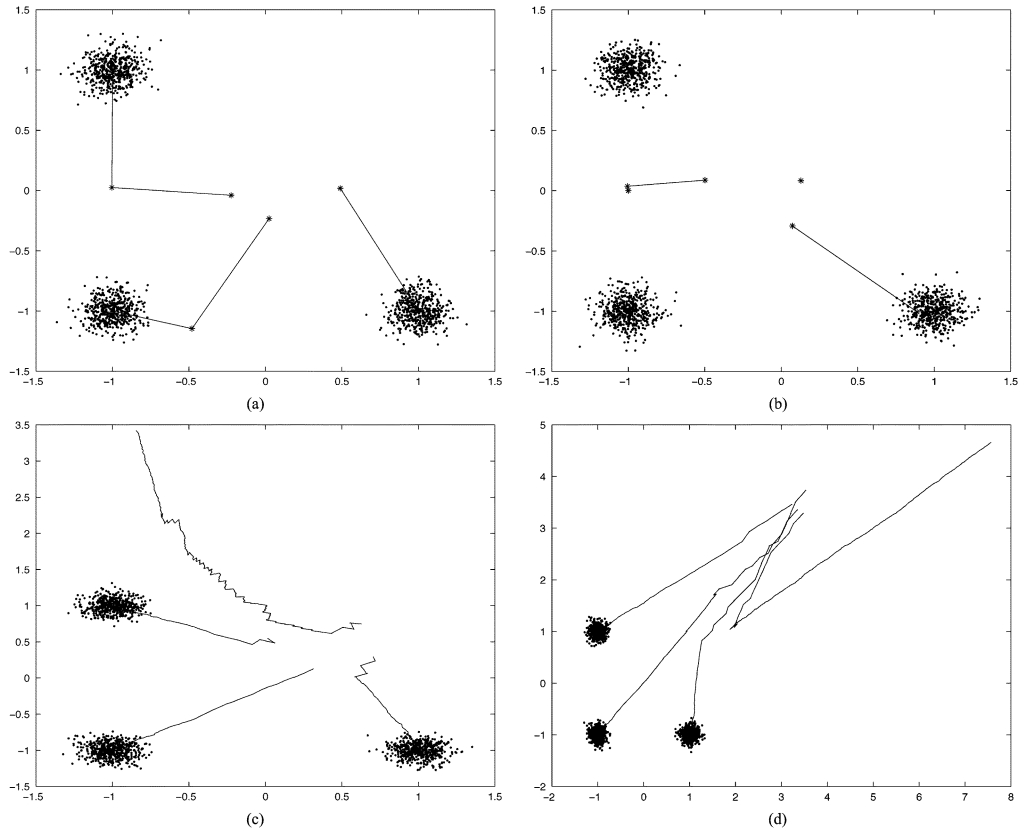

Fig. 5. Clustering results with k-Means and original RPCL. (a) $\mathrm{k-Means}(k=3)$ succeeds in finding three clusters. (b) $\mathrm{k-Means}(k=3)$ fails in finding three clusters. (c) Original $\mathrm{RPCL}(k=4)$ succeeds in finding three clusters with initialization domain [0.0, 1.0]. (d) Original $\mathrm{RPCL}(k=4)$ succeeds in finding three clusters with initialization domain [3.0, 4.0].

The approach introduced above is called original RPCL. It only considers the center of clusters and selects the winner according to distance measure. The riginal RPCL was extended to cluster data sets with complicated shapes via the finite mixture modeling. When the distribution for each cluster is assumed to be a Gaussian $\mathcal{N}(m_i, \Sigma_i)$, an adaptive form of RPCL is proposed as follows [21]:

$$\begin{aligned} c &= \arg\min_j d_j \\ r &= \arg\min_{j\neq c} d_j \\ d_j &= (\vec{x}-\vec{m}_j)'\Sigma_j^{-1}(\vec{x}-\vec{m}_j) \\ &\quad + \ln|\Sigma_j| - 2\ln\gamma_j. \end{aligned} \tag{4}$$

Updating on parameters $\vec{m}_j$ becomes exactly the same with (3). The update of the covariance matrix Σ_j is as follows:

$$\begin{aligned} \Sigma_c^{(t+1)} &= (1-\alpha_c)\Sigma_c^{(t)} + \alpha_c(\vec{x}-\vec{m}_c)(\vec{x}-\vec{m}_c)' \\ \Sigma_j^{(t+1)} &= \Sigma_j^{(t)}, \qquad j \neq c. \end{aligned} \tag{5}$$

C. Comparison

The test samples are comprised of three clusters, and each cluster has 500 points from a Gaussian distribution, their means are at $(-1.0, 1.0), (-1.0, -1.0)$, and $(1.0, -1.0)$ respectively.

For k-Means, the preset number is 3, and the initial center is generated randomly from the domain $[-0.5, 0.5]$. In order to show the automatic number selection ability, the initial number of clusters for RPCL is set to 4, and the initial center is generated randomly from two domain, [3.0, 4.0] and [0.0, 1.0], the learning and de-learning rate are set to $\alpha_c = 0.05, \alpha_r = 0.002$, respectively.

Even when the right cluster number is given, k-Means cannot guarantee the correct clustering results. From Fig. 5(b), we can see that clustering results of k-Means is very dependent on initial positions. While for RPCL, clustering succeeds on all the independent runs by automatically selecting correct number, even if the initialization domain changes [see Fig. 5(c) and (d)]. Thus, RPCL not only has the unique automatic number selection ability, but also shows better reliability.

IV. Clustering and Learning Gaussian Distribution Algorithms

Two new EDAs that take advantage of clustering are presented in this section. The first combines original RPCL with $\mathrm{EGNA_{BGe}}$, so it is called the Clustering and Estimation of Gaussian Network Algorithm based on BGe metric $(\mathrm{CEGNA_{BGe}})$. The other uses adaptive RPCL as both a clustering and estimation approach, and is called the Clustering and Estimation of Gaussian Distribution Algorithm (CEGDA). Both algorithms regard the data as a random sample from a finite mixture of multivariate Gaussian distribution, and multiple interdependencies between variables are considered in both of them.

A. $\mathrm{CEGNA_{BGe}}$

The design of this algorithm is to cluster the selected individuals into clusters by original RPCL approach, and for each single cluster, $\mathrm{EGNA_{BGe}}$[2] is used to estimate the distribution. Because of the use of RPCL, prior knowledge is not required, and the actual number will be decided during the clustering process. To be exact, at the beginning of clustering, a small number is assigned as the initial assumed number of clusters, if any extra cluster is found at the end of clustering, the clustering process stops and the left clusters are estimated; otherwise the clustering process is repeated with the cluster number doubled, until the stopping criterion is met. Here, an extra cluster is a cluster whose size is less than a predefined threshold value. For example, if some outliers exist in the data set, RPCL will allocate a unit to it, and the size of this cluster is 1. But this cluster will not be retained, but discarded as noise information. The new individuals are simulated based on each estimated distribution, and the number of solutions from a certain cluster is proportional to its average fitness.

The framework of $\mathrm{CEGNA_{BGe}}$ is as follows.

1) Initialize a population of N individuals randomly.
2) Select $Se \leq N$ individuals according to some selection scheme.
3) Cluster the selected individuals into k clusters by original RPCL.
4) Estimate the distribution $P_i(x), i = 1, \ldots, k$ by $\mathrm{EGNA_{BGe}}$.
5) Sample $N_i (\sum_{i=1}^{k} N_i = N)$ new individuals based on $P_i(x)$ to replace the old population, N_i is proportional to the average fitness of each cluster.
6) Stop if some stopping criterion is reached, go to 2 otherwise.

B. CEGDA

The joint density function of a multivariate Gaussian distributions can be written as a product of conditional density functions, each of which belongs to an independent Gaussian distribution. That is

$$p(x) = \prod_{i=1}^{n} \mathcal{N}\left(\mu_i + \sum_{j=1}^{i-1} b_{ji}(x_j - \mu_j), \sigma_i\right) \tag{6}$$

where $\mathcal{N}(\mu, \sigma)$ represents a univariate Gaussian density function with mean μ and standard deviation σ. μ_i in (6) is the unconditional mean vector of x_i, σ_i is the conditional standard deviation of x_i given values for $x_1, x_2, \ldots, x_{i-1}$, and b_{ji} is a linear coefficient reflecting the strength of the relationship between x_i and x_j.

Shachter and Kenley [23] described the general transformation from $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ and $\{b_{ji} \mid j < i\}$ to the precision matrix W of Gaussian distribution, which is the inverse of covariance matrix. They use the following recursive formula in which $W(i)$ denotes the $i \times i$ upper left submatrix of $W, \vec{b_i}$ denotes the $(i-1)$-dimensional column vector $(b_{1,i}, \ldots, b_{i-1,i})$, and $\vec{b_i}'$ denotes the transposed vector

$$W(i+1) = \begin{pmatrix} W(i) + \frac{\vec{b}_{i+1}\vec{b}'_{i+1}}{\sigma_{i+1}} & -\frac{\vec{b}_{i+1}}{\sigma_{i+1}} \\ -\frac{\vec{b}'_{i+1}}{\sigma_{i+1}} & \frac{1}{\sigma_{i+1}} \end{pmatrix}. \tag{7}$$

This transformation makes it possible to construct a new algorithm. Since adaptive RPCL updates both the mean and the covariance matrix at each step of clustering, and the factorization form of Gaussian can be obtained from mean and covariance matrix using the transformation formula, it means adaptive RPCL can finish clustering and estimation at the same time. A method known as conditioning simulation method is used in both $\mathrm{CEGNA_{BGe}}$ and CEGDA. Based on (6), x_1 is generated first, and then generate x_2 given x_1, and so on. In summary, the framework of CEGDA is as follows.

1) Initialize a population of N individuals randomly.
2) Select $\mathrm{Se} \leq N$ individuals according to some selection scheme.
3) Cluster the selected individuals into k clusters and return the estimation of the distributions of each cluster $P_i(x), i = 1, \ldots, k$ by adaptive RPCL.
4) Sample N_i $(\sum_{i=1}^{k} N_i = N)$ new individuals based on $P_i(x)$ to replace the old population, N_i is proportional to the average fitness of each cluster.
5) Stop if some stopping criterion is reached, otherwise go to 2.

V. Experimental Analysis

In this section, experimental results on a set of functions are presented to show the performance of $\mathrm{CEGNA_{BGe}}$ and CEGDA in all aspects, the strongpoint as well as the weakness.

A. Experimental Design

Six test functions are chosen purposely, two of them are unimodal functions, and the others multimodal.

Sphere [see Fig. 6(a)]

$$F(\vec{x}) = \sum_{i=1}^{n} x_i^2. \tag{8}$$

SumCan [see Fig. 6(b)]

$$\begin{aligned} F(\vec{x}) &= 1/\left(10^{-5} + \sum_{i=1}^{n} |y_i|\right) \\ y_1 &= x_1 \\ y_i &= x_i + y_{i-1} \quad i \geq 2. \end{aligned} \tag{9}$$

TwoPeaks [see Fig. 6(c)]

$$F(\vec{x}) = \sum_{i=1}^{2} \alpha_i \mathcal{N}(\vec{x}, \mu_i, \Sigma_i) \tag{10}$$

[2]For details of the implementation of $\mathrm{EGNA_{BGe}}$, refer to [12]

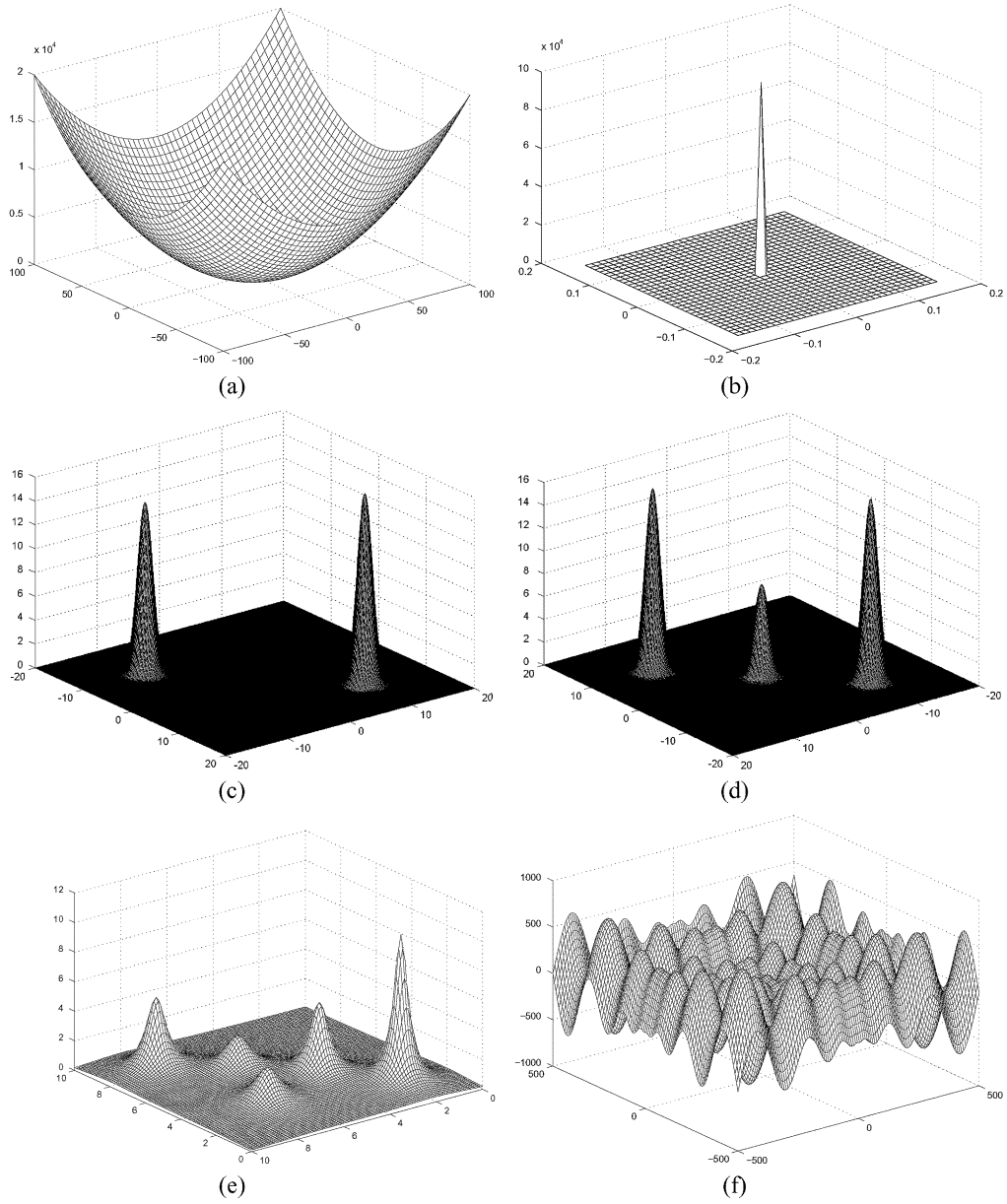

Fig. 6. Plots of the problems (dimension = 2) to be optimized by continuous EDAs and CEP. (a) Sphere model. (b) SumCan model. (c) TwoPeaks model. (d) ThreePeaks model. (e) Shekel model with five local optima. (f) Schwefel model.

where $\mathcal{N}(\vec{x}, \mu_i, \Sigma_i)$ is a multivariate normal distribution, which has n-dimensional mean vector μ and $n \times n$ covariance matrix Σ. $\alpha_1 = 1000, \alpha_2 = 900, \mu_1 = (-10, \ldots, -10), \mu_2 = (10, \ldots, 10), \Sigma_i (i = 1, 2)$ are diagonal matrices with all the diagonal elements equalling 1.

ThreePeaks [see Fig. 6(d)]

$$F(\vec{x}) = \sum_{i=1}^{3} \alpha_i \mathcal{N}(x, \mu_i, \Sigma_i) \tag{11}$$

where the same settings with TwoPeaks model plus $\alpha_3 = 500$ and $\mu_3 = (0, \ldots, 0)$.

Shekel[3] [see Fig. 6(e)]

$$F(\vec{x}) = \sum_{i=1}^{n} [(\vec{x} - \vec{a_i})(\vec{x} - \vec{a_i})^T + \vec{c_i}]^{-1}$$

i	$a_{ij}, j = 1, \ldots, 4$	c_i
1	2 2 2 2	0.1
2	4 4 4 4	0.2
3	8 8 8 8	0.2
4	6 6 6 6	0.4
5	3 7 3 7	0.4

. (12)

[3] n is set to 5 in the experiment.

TABLE I
SETTINGS FOR ALL THE TEST FUNCTIONS

	Dimension	Domain	Type	Optimum
Sphere	30	[-100, 100]	Min.	0
SumCan	10	[-0.16, 0.16]	Max.	10^5
TwoPeaks	5	[-100, 100]	Max.	10.1053
ThreePeaks	5	[-100, 100]	Max.	10.1053
Shekel(n=5)	4	[0, 10]	Max.	10.1033
Shekel(n=5)	30	[0, 10]	Max.	10.0134
Schwefel	30	[-500, 500]	Min.	-12569.5

TABLE II
EXPERIMENTAL SETTINGS FOR ALL THE TEST ALGORITHMS

	Population	Selection	α_c	α_r
CEP	100			
$\mathrm{UMDA_c}$	1000	500		
$\mathrm{EGNA_{BGe}}$	1000	500		
$\mathrm{CEGNA_{BGe}}$	2000	500	0.05	0.002
CEGDA	2000	500	0.05	0.002

Schwefel [see Fig. 6(f)]

$$F(\vec{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}). \tag{13}$$

The particular two-dimensional case for each function is shown in Fig. 6, and the settings are in Table I.

B. Algorithms Settings

Four EDA approaches, which are $\mathrm{UMDA_c}$ [12], $\mathrm{EGNA_{BGe}}$ [12], $\mathrm{CEGNA_{BGe}}$ and CEGDA, and classical EP (CEP) [24] are used on the test functions for comparison purpose.

For each algorithm, the settings are unchanged through all experiments. All mean results were averaged over 30 independent runs. The initial population is generated uniformly at random in the domain specified in the description of each function. The detailed settings are listed in Table II.

In Table II, *Population* is the number of individuals in each generation, *Selection* is the number of promising solutions selected from all the individuals in the population based on some predefined selection scheme, and α_c and α_r are initial learning and de-learning rates used with RPCL.

C. Results and Analysis

The experimental results are summarized in Tables III–IX. All results have been averaged over 30 independent runs, and the maximal evaluation number for unimodal and multimodal functions are 2×10^5 and 4×10^5, respectively. All the EDA algorithms use truncate selection, while CEP uses tournament selection with tournament size equals to 10. Specially for $\mathrm{CEGNA_{BGe}}$ and CEGDA, the initial cluster number is set to 2. Also, the threshold value used with RPCL for deciding which cluster can be discarded is set to 5% of the size of whole data set.

The analysis is divided into four parts based on different points of view.

TABLE III
EXPERIMENTAL RESULTS FOR THE SPHERE FUNCTION

	Gen	Best	Mean	Std
CEP	2000	2.76e-007	1.93e-004	4.58e-004
$\mathrm{UMDA_c}$	200	1.88e-016	3.24e-016	5.59e-017
$\mathrm{EGNA_{BGe}}$	200	5.86e-010	1.20e-009	3.40e-009
$\mathrm{CEGNA_{BGe}}$	100	7.27e-010	1.19e-008	2.99e-008
CEGDA	100	3.38e-008	3.41e-006	8.40e-007

TABLE IV
EXPERIMENTAL RESULTS FOR THE SUMCAN FUNCTION

	Gen	Best	Mean	Std
CEP	2000	3.87988	2.39695	0.503663
$\mathrm{UMDA_c}$	200	698.72	221.771	116.101
$\mathrm{EGNA_{BGe}}$	200	100000	100000	0
$\mathrm{CEGNA_{BGe}}$	100	24.3826	13.4422	4.06832
CEGDA	100	99834.5	99748.1	63.2197

TABLE V
EXPERIMENTAL RESULTS FOR THE TWOPEAKS FUNCTION

	Gen	Best	Mean	Std
CEP	4000	0	0	0
$\mathrm{UMDA_c}$	400	10.1053	9.6327	0.1073
$\mathrm{EGNA_{BGe}}$	400	10.1053	9.8324	0.0828
$\mathrm{CEGNA_{BGe}}$	200	10.1053	10.1053	3.55e-015
CEGDA	200	10.1053	10.0999	5.92e-003

TABLE VI
EXPERIMENTAL RESULTS FOR THE THREEPEAKS FUNCTION

	Gen	Best	Mean	Std
CEP	4000	9.18e-107	3.06e-108	1.64e-107
$\mathrm{UMDA_c}$	400	5.05266	5.05266	8.88e-016
$\mathrm{EGNA_{BGe}}$	400	5.05266	5.05266	8.88e-016
$\mathrm{CEGNA_{BGe}}$	200	10.1053	10.1053	3.55e-015
CEGDA	200	10.1053	10.1048	7.99e-004

TABLE VII
EXPERIMENTAL RESULTS FOR THE SHEKEL FUNCTION WITH FIVE LOCAL OPTIMA ($\mathrm{D} = 4$)

	Gen	Best	Mean	Std
CEP	4000	10.1033	6.7407	2.6372
$\mathrm{UMDA_c}$	400	5.1877	4.7331	0.7406
$\mathrm{EGNA_{BGe}}$	400	8.2036	4.9691	0.7786
$\mathrm{CEGNA_{BGe}}$	200	10.1033	10.1033	8.8818e-015
CEGDA	200	10.1033	10.1033	8.8818e-015

1) Easy Unimodal Problem: Sphere model has only one global optimum without any local optimum and the variables are independent of each other. Basically it appears rather easy for nearly any evolutionary algorithms. From Table III, it appears that $\mathrm{CEGNA_{BGe}}$ and CEGDA can both perform well on the Sphere function. Actually, because of simplicity, all the test algorithms can find a very good solution at the end. It can

TABLE VIII
EXPERIMENTAL RESULTS FOR THE SHEKEL FUNCTION WITH FIVE LOCAL OPTIMA (D = 30)

	Gen	Best	Mean	Std
CEP	4000	10.0134	5.8327	2.8914
$\mathrm{UMDA_c}$	400	5.1325	4.3108	0.9326
$\mathrm{EGNA_{BGe}}$	400	6.3394	4.1025	0.8823
$\mathrm{CEGNA_{BGe}}$	200	10.0134	10.0134	8.8818e-015
CEGDA	200	10.0134	10.0134	8.8818e-015

TABLE IX
EXPERIMENTAL RESULTS FOR THE SCHWEFEL FUNCTION

	Gen	Best	Mean	Std
CEP	4000	-9470.28	-8144.09	653.102
$\mathrm{UMDA_c}$	400	-5928.24	-5424.81	202.437
$\mathrm{EGNA_{BGe}}$	400	-12569.5	-12276	189.539
$\mathrm{CEGNA_{BGe}}$	200	-10773.1	-6760.35	2624.33
CEGDA	200	-8712.31	-5922.54	1893.51

be noticed that among the EDAs, the algorithms with single Gaussian distribution assumption outperform the new proposed algorithms using clustering techniques. Specifically, $\mathrm{UMDA_c}$, which does not consider any interdependencies, performs far better than the others. The reason is that for unimodal functions, the single Gaussian distribution assumption is more suitable than the mixture of Gaussian model; moreover, this particular function nicely accords with the no interdependencies assumption. Also, all the algorithms based on distribution estimation have better performance than CEP, which mainly depends on random search.

2) Problem With Strong Interdependencies: The SumCan function is a unimodal function with very strong interdependencies between the variables, and was used in [12] to show the advantage of algorithm $\mathrm{EGNA_{BGe}}$ over $\mathrm{UMDA_c}$ and $\mathrm{MIMIC_c}$.

All the EDA algorithms are better than CEP in this specific case, since it is very hard for CEP to find a good solution when fitness value is approximately equal to zero at most points in the search space. Inside the EDA family, $\mathrm{UMDA_c}$ performs very badly because it ignores any interdependency information. The most interesting result is that even if $\mathrm{CEGNA_{BGe}}$ and CEGDA do not assume any independency, they are both outperformed by $\mathrm{EGNA_{BGe}}$, especially $\mathrm{CEGNA_{BGe}}$ appears the worst among all the EDA approaches. The reason is that although $\mathrm{EGNA_{BGe}}$ and $\mathrm{CEGNA_{BGe}}$ use an identical distribution estimation procedure, the data set for estimation is totally different. $\mathrm{EGNA_{BGe}}$ uses the whole set of selected individuals, while $\mathrm{CEGNA_{BGe}}$ clusters first and estimates the distribution for each cluster thereafter. With dimension size equal to 10, it is impossible for all the selected individuals to be selected into one cluster by original RPCL which only use the distance as the measure. As a result, each cluster only comprises of individuals from a very small space, and the new individuals will inevitably be trapped in this area, the whole optimization process will converge undesirably. As for CEGDA, despite it that also uses clustering, it behaves notably better than $\mathrm{CEGNA_{BGe}}$. In fact, the best value from CEGDA is very close to the global optimum, which indicates that CEGDA really has captured the interdependencies between the variables. The reason why it cannot find the global optimum may be that the estimated Gaussian distribution, which makes use of adaptive RPCL, is a little less accurate than that of the other method, such as $\mathrm{EGNA_{BGe}}$. The difference between CEGDA and $\mathrm{CEGNA_{BGe}}$ comes from the use of adaptive RPCL in CEGDA. In adaptive RPCL, the pure distance measure is substituted by the density measure, which enables each cluster to hold distance data. So in CEGDA, a lot of new individuals farther from the current mean can be generated, which prevents the premature happening like $\mathrm{CEGNA_{BGe}}$.

3) Multimodal Problems With a Few Local Optima: In this part, the results from three test functions are discussed. These three functions, TwoPeaks, ThreePeaks, and Shekel, are all multimodal functions. Because one important motivation of constructing these two new algorithms is to take advantage of clustering technique to overcome the difficulty in dealing with multimodal functions. Thus, the performance on these three problems can evaluate if this goal has achieved. Compared with $\mathrm{UMDA_c}$ and $\mathrm{EGNA_{BGe}}$, the two new EDAs show better performance on all three problems. $\mathrm{CEGNA_{BGe}}$ can reach the global optimum in every run, and although CEGDA cannot guarantee to reach the global optimum on ThreePeaks problem every time, the error is infinitesimal. This little difference between $\mathrm{CEGNA_{BGe}}$ and CEGDA once again gives some empirical proof to support the deduction based on the results on the SumCan function, which is the estimation by adaptive RPCL is a little bit less accurate than that from $\mathrm{EGNA_{BGe}}$.

Besides correctly finding one global optimum, the automatic number selection feature of RPCL can make it possible to find the exact number of global optima provided the solving problem only has a few local optima. In order to illustrate the autonomic number selection ability in a clearer way, the optimization process for the 2-dimensional ThreePeaks function by $\mathrm{CEGNA_{BGe}}$ is traced, and the distributions of selected individuals at different stage are shown in Fig. 7. At generation 1, just after the initialization, the selected individuals are located in the space rather uniformly. Until generation 5, the population has begun to gather into three groups, two around global optima separately and the middle one around the local optimum. Because at each generation, new individuals are generated in proportion to the average fitness of each cluster, more individuals around global optimum are simulated and the size of the cluster representing local optimum decreases step by step. At generation 10, it can be noticed that only two clusters are left, each of which contains a global optimum. Similar processes occur when optimizing TwoPeaks and Shekel functions. This process fully indicates that making use of the automatic number selection ability of RPCL, $\mathrm{CEGNA_{BGe}}$ and CEGDA have the ability to detect the number of global optima automatically if the number of local optima is small.

CEP performs poorly on TwoPeaks and ThreePeaks problems due to the same reason when applied to SumCan function, which is the needle-in-stack alike landscape. For the Shekel function, CEP can find the global optimum in 13 $(\mathrm{d} = 4)$ and 5 $(\mathrm{d} = 30)$ runs out of 30, and some local optima are found in the other runs.

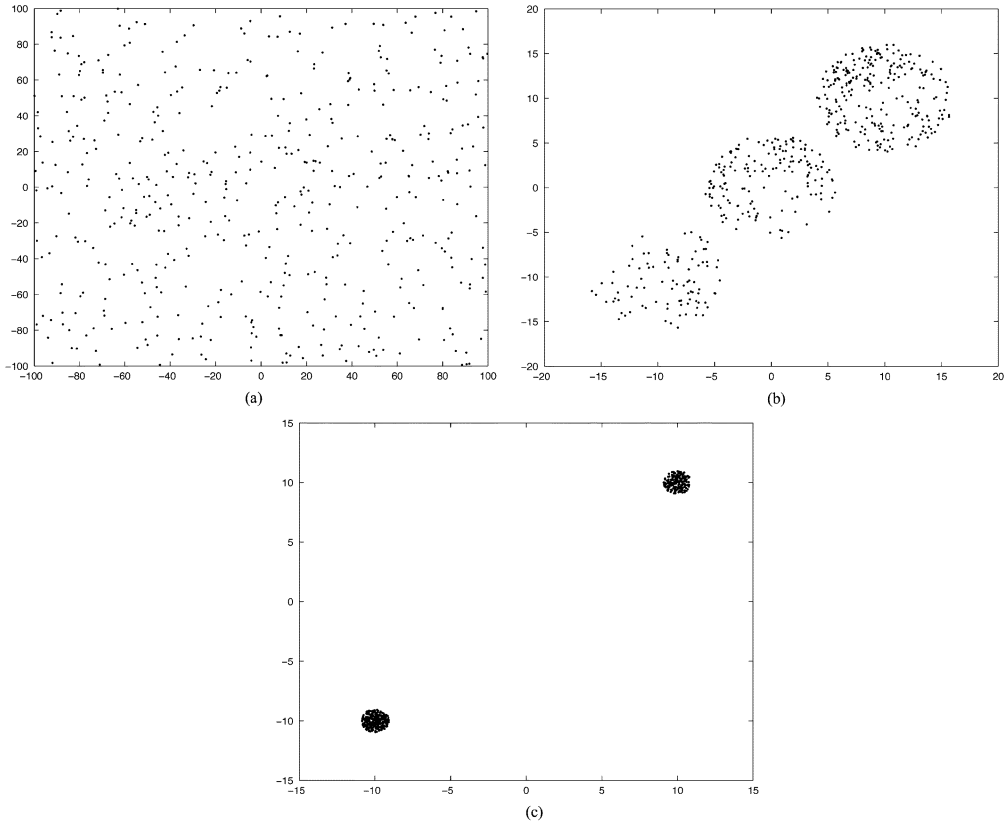

Fig. 7. Demonstrate how $\mathrm{CEGNA_{BGe}}$ can automatically detect the number of global optima of ThreePeaks function. (a) Distribution of selected 500 individuals at generation 1. (b) Distribution of selected 500 individuals at generation 5. (c) Distribution of selected 500 individuals at generation 10.

4) Multimodal Problems With Many Optima: Schwefel function contains many local optima. The results show that $\mathrm{UMDA_c}$ is the worst, $\mathrm{EGNA_{BGe}}$ is the best and the other three hold comparable behavior. Due to the large number of local optima, under practical conditions, the global optimum can hardly be detected, every selected cluster normally contains data from the neighborhood of different local optima. Because EDAs use estimated distribution to guide the search, and an accurate distribution estimation needs enough samples, when the number of local optima is very large, the population size has to be very large in order to capture every local optimum, which is impractical.

5) Effect of Automatic Number Selection: Finally, the algorithm presented by Pelican and Goldberg [17] is used here to examine the effect of automatic number selection to the result of optimization. This algorithm is equivalent with $\mathrm{k-Means+UMDA_c}$ (called $\mathrm{CUMDA_c}$ in the following context). Apply the new EDAs accompanying with $\mathrm{CUMDA_c}$ to the Shekel function $(\mathrm{d}=30)$, and the results are summarized in Table X. The cluster number provided to k-Means equals 4, 5, 6, 10, respectively, the population size and selection size are 2000 and 1000.

It appears that even if you give a larger number than the number of local optima, it cannot guarantee that the global optimum can be found. The reason is that the clustering result of k-Means relies on the initialization, which means using a larger number can produce the same clustering result as the result from using a smaller number. Overall, the ability of automatic number selection provided by RPCL shows significant advantage over the previous used clustering techniques.

TABLE X
EXPERIMENTAL RESULTS FOR EXAMINING THE EFFECT OF AUTOMATIC NUMBER SELECTION

	Gen	Best	Mean	Std
$\mathrm{CUMDA_c}$(k=4)	200	10.0134	4.8109	2.5296
$\mathrm{CUMDA_c}$(k=5)	200	10.0134	6.5473	1.2367
$\mathrm{CUMDA_c}$(k=6)	200	10.0134	7.2891	0.8324
$\mathrm{CUMDA_c}$(k=10)	200	10.0134	9.0107	0.2653
$\mathrm{CEGNA_{BGe}}$	200	10.0134	10.0134	8.8818e-015
CEGDA	200	10.0134	10.0134	8.8818e-015

VI. CONCLUSION

This paper proposes two new EDAs for continuous optimization, $\mathrm{CEGNA_{BGe}}$ and CEGDA, and analyzes the strongpoint and weakness of them through a set of experiments. The improvement of the new algorithms over the existing ones is twofold: the first improvement is by incorporating clustering technique, the new algorithms can solve multimodal functions with a few local optima very successfully, while those EDAs using the single Gaussian distribution assumption cannot solve effectively; the second advantage is by making use of the automatic number selection ability of RPCL, new algorithms for the first time in EDAs have the ability to detect the number of global optima automatically, no prior knowledge is needed for the algorithms to work. At the same time, it is also found that if a multimodal function has too many local optima, it appears hard for the new algorithms to conquer.

Besides these common features, $\mathrm{CEGNA}_{\mathrm{BGe}}$ and CEGDA also have some differences between them. $\mathrm{CEGNA}_{\mathrm{BGe}}$ is easy to cause the optimization process to converge prematurely under some particular conditions (e.g., SumCan function). CEGDA does not have this deficiency, but as a result of doing the clustering and estimation concurrently using adaptive RPCL, when facing a data set with strong interdependencies between the variables, its estimated result is slightly less accurate compared with the other estimation model.

Up to now, most research works use Gaussian distribution as model to construct new algorithms in EDA category. The other distributions can be considered in the future work.

REFERENCES

[1] S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.

[2] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evolut. Comput.*, vol. 5, pp. 303–346, 1998.

[3] G. R. Harik, E. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," in *Proc. Int. Conf. Evolutionary Computation 1998 (ICEC'98)*, 1998, pp. 523–528.

[4] J. S. D. Bonet, C. L. Isbell, and P. Viola, "MIMIC: Finding optima by estimating probability densities," *Adv. Neural Inform. Process. Syst.*, vol. 9, p. 424, 1997.

[5] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," in *Proc. 14th Int. Conf. Machine Learning*, 1997, pp. 30–38.

[6] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," *Adv. Soft Comput.—Eng. Design and Manuf.*, pp. 521–535, 1999.

[7] R. Etxeberria and P. Larrañaga, "Global optimization with bayesian networks," in *Symp. Artificial Intelligence CIMAF'99—Special Session on Distributions and Evolutionary Optimization*, 1999, pp. 332–339.

[8] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "Boa: The Bayesian optimization algorithm," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO-99)*, vol. I, 1999, pp. 525–532.

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

[10] S. Rudlof and M. Köppen, "Stochastic hill climbing by vectors of normal distributions," in *Proc. 1st Online Workshop on Soft Computing (WSC1)*, Nagoya, Japan, 1996.

[11] I. Servet, L. Trave-Massuyes, and D. Stern, "Telephone network traficc overloading diagnosis and evolutionary techniques," in *Proc. 3rd European Conf. Artificial Evolution (AE'97)*, 1997, pp. 137–144.

[12] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña, "Optimization by Learning and Simulation of Bayesian and Gaussian Networks," Dept. Comput. Sci. Artific. Intell., Univ. Basque Country, Tech. Rep. EHU-KZAA-IK-4/99, 1999.

[13] ——, "Optimization in continuous domains by learning and simulation of Gaussian networks," *Proc. 2000 Genetic and Evolutionary Computation Conf. Workshop Program*, pp. 201–204, 2000.

[14] P. A. N. Bosnian and D. Thierens, "Expanding from discrete to continuous estimation of distribution algorithms: The IDEA," in *Parallel Problem Solving from Nature—PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. New York: Springer-Verlag, 2000, pp. 767–776.

[15] ——, "Exploiting gradient information in continuous iterated density estimation evolutionary algorithms," in *Proc. 13th Belgium-Netherlands Artificial Intelligence Conf. (BNAIC-2001)*, B. Kröse, M. de Rijke, G. Schreiber, and M. van Someren, Eds., 2001, pp. 69–76.

[16] M. Gallagher, M. Frean, and T. Downs, "Real-valued evolutionary optimization using a flexible probability density estimator," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO'99)*, vol. 1, 1999, pp. 840–846.

[17] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Proc. Parallel Problem Solving from Nature—PPSN VI*, Paris, France, 2000, pp. 385–394.

[18] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA: Kluwer, 2001, ch. 4, pp. 101–127.

[19] 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.

[20] L. Xu, A. Krżyzak, and E. Oja, "Rival penalized competitive learning for clustering analysis, rbf net, and curve detection," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 636–649, Jul. 1993.

[21] L. Xu, "Rival penalized competitive learning, finite mixture, and multisets clustering," in *Proc. Int. Joint Conf. Neural Networks*, vol. II, 1997, pp. 2525–2530.

[22] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evolut. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[23] R. Shachter and C. Kenley, "Gaussian influence diagrams," *Manag. Sci.*, vol. 35, pp. 527–550, 1989.

[24] X. Yao, Y. Liu, and G. Liu, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

Qiang Lu received the B.Sc. degree in applied mathematics and the M.Sc. degree in operations research from Tsinghua University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2004.

He is currently a Research Fellow in the School of Physics and Astronomy, University of Birmingham. His research interest is in evolutionary computation.

Xin Yao (M'91–SM'96–F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc.degree from the North China Institute of Computing Technology (NCI), Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is currently a Professor of Computer Science in the Natural Computation Group and Director of the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA) at the University of Birmingham, Birmingham, U.K. He is also a Distinguished Visiting Professor at the University of Science and Technology of China, Hefei, and a Visiting Professor at Nanjing University of Aeronautics and Astronautics, Nanjing, China, the Xidian University, Xi'an, China, and the Northeast Normal University, Changchun, China. His major research interests include evolutionary computation, neural network ensembles, global optimization, computational time complexity and data mining.

Dr. Yao is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, an Associate Editor or Editorial Board Member of ten other international journals, and the past Chair of the IEEE Neural Network Society's Technical Committee on Evolutionary Computation. He is the recipient of the 2001 IEEE Donald G. Fink Prize Paper Award and has given more than 27 invited keynote/plenary speeches at various international conferences.