Big Data and Information Analytics
©American Institute of Mathematical Sciences
Volume **2**, Number **1**, January **2017**
doi:10.3934/bdia.2017010
pp. **77–85**

A CLUSTERING BASED MATE SELECTION FOR EVOLUTIONARY OPTIMIZATION

Jinyuan Zhang, Aimin Zhou*, and Guixu Zhang

Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology
East China Normal University,Shanghai, 200062, China

Hu Zhang

Beijing Electro-Mechanical Engineering Institute
Beijing, 100074, China

Abstract. The mate selection plays a key role in natural evolution process. Although a variety of mating strategies have been proposed in the community of evolutionary computation, the importance of mate selection has been ignored. In this paper, we propose a *clustering based mate selection (CMS)* strategy for *evolutionary algorithms (EAs)*. In CMS, the population is partitioned into clusters and only the solutions in the same cluster are chosen for offspring reproduction. Instead of doing a whole new clustering process in each EA generation, the clustering iteration process is combined with the evolution iteration process. The combination of clustering and evolving processes benefits EAs by saving the cost to discover the population structure. To demonstrate this idea, a CMS utilizing the k-means clustering method is proposed and applied to a state-of-the-art EA. The experimental results show that the CMS strategy is promising to improve the performance of the EA.

1. **Introduction.** In this paper, we consider the following continuous *global optimization problem*.

$$\min f(x) \text{ s.t } x \in [a_i, b_i]^n \tag{1}$$

where $x = (x_1, x_2, \cdots, x_n) \in R^n$ is a decision variable vector; $[a_i, b_i]^n$ defines the feasible range of the decision space and $a_i \leq x_i \leq b_i$ for $i = 1, 2, \cdots, n$; and $f : R^n \to R$ is the objective function.

The *evolutionary algorithm (EA)* is a type of heuristic optimization method, which is inspired by the natural evolution process [1]. It has become a major method to tackle (1). The major components in a general EA include a reproduction operator, and a selection operator. There is a key operation in natural evolution, named as *mate selection*, which chooses mating pairs or groups for breeding and plays a key role in sexual propagation. In EAs, a proper mate selection can also control the population convergence and diversity efficiently [6, 12, 13]. In the last

2010 *Mathematics Subject Classification.* 78M32.

Key words and phrases. Evolutionary algorithm, mate selection, clustering.

This work is supported by the National Natural Science Foundation of China under Grant No. 61673180 and 61703382, the Shanghai Clearing House under the project of 'artificial intelligence methods for complex 0-1 financial optimization', and the Open Project of Shanghai Key Laboratory of Trustworthy Computing under Grant No. 07dz22304201507.

* Corresponding author: A. Zhou.

decades some mating strategies have been proposed [16], including random mating, roulette wheel selection, truncate selection, tournament selection, gender based selection [7, 15, 19], niche based selection [2], dissociative selection [3, 4], and some other methods [5, 14, 18]. Although these mating strategies have been proposed, it has not attracted much attention in the community of evolutionary computation [16]. The major reasons might be that (a) most of existing mating strategies need some problem specific control parameters or are computationally expensive, and (b) some widely used EAs work well by randomly choosing mating pairs. In this paper, we shall demonstrate that existing EAs can be improved by using properly designed mating strategies.

Statistical and machine learning (SML) techniques aim to extract information from data sets and transform it into an understandable pattern or structure for further use [8]. It is arguable that in EAs, an individual can be regarded as a training example, and its corresponding fitness value be a label. From the viewpoint of SML, the population of an EA forms a training data set. Therefore, SML techniques can be naturally applied to EAs to extract population information and guide the search. Some algorithms, such as estimation of distribution algorithms [10], and surrogate assisted evolutionary algorithms [9], are along this direction. Basically SML techniques are computationally expensive comparing to general EAs, which limits their usages. How to use SML techniques in EAs more efficiently is still an open question.

In this paper, we present a new way to combine SML methods with EAs. The basic idea is to iteratively call SML training step and EA evolving step. In the SML step, the obtained population is utilized to train a model that captures the population structure, and then in the EA step, the population structure information extracted in the SML step is used to guide the search. The combination of the SML iteration process and the EA iteration process can find and refine the population structure information and thus save the SML cost. In multi-objective evolutionary optimization, there are some works with similar idea [20]. However for scalar-objective optimization, this strategy is still new. Based on this idea, this paper proposes a *clustering based mate selection (CMS)* operator for EAs. In CMS, the population is partitioned into classes in each generation, and only the solutions in the same class are allowed to mate with each other. A CMS utilizing the k-means [11] clustering method is proposed and applied to a state-of-the-art EA to show its advantages.

The rest of the paper is organized as follows. Section 2 presents the proposed CMS strategy. An EA integrated CMS is introduced in detail as well. Section 3 compares the proposed CMS strategy with some other mating strategies, and studies the influence of the control parameters. Finally, the paper is concluded in Section 4.

2. Clustering based mate selection.

A major challenge by applying SML techniques in EAs is on the high computational cost. This section introduces a *clustering based mate selection (CMS)* to address the challenge. The basic idea is to combine an EA with an iterative clustering method together. Take the k-means method as an example. In each generation (iteration) of the combined process, the clustering process uses the EA population to assign points and update cluster centers; and then based on the population partition, an EA chooses the parents in the same cluster to generate new trails solutions. It should be noted that the CMS assisted EA does not implement a clustering method in each generation. Instead it combines the

clustering iteration with the EA iteration, and only one clustering iteration is implemented in each generation. Actually, the clustering process is only implemented several times sequentially along with the EA process. By this way, the computational cost is saved up. The major components of an iterative clustering method and an EA are combined in the CMS assisted EA (CMS-EA for short). A restart checking component is added to reinitialize the clustering. A major reason is to avoid getting into local optima in clustering.

In this paper, we use the CMS strategy to improve the performance of the *composite differential evolution (CoDE)* algorithm [21]. In CoDE, each solution produces three candidate offspring solutions by using three reproduction operators with randomly selected three control parameters, and chooses the best candidate as the offspring solution for updating. More details of CoDE can be found in [21]. The k-means clustering method is used to partition the population. In the following, we give the framework of the proposed approach, named as CMS-CoDE.

---

1 Randomly initialize a population $P = (x^1, x^2, \cdots, x^N)$, and set generation count $g = 0$.

2 If $mod(g, G) = 0$, initialize the cluster centers $m^1, m^2, \cdots, m^K$, and set K empty clusters $C^1, C^2, \cdots, C^K$.

3 For each solution x^i, $(i = 1, 2, \cdots, N)$, assign it to the k-th cluster C^k which satisfies

$$k = \arg \min_{j=1,2,\cdots,K} dis(x^i, m^j),$$

where $dis(a, b)$ is the Euclidean distance between a and b.

4 For each cluster C^k, $(k = 1, 2, \cdots, K)$, update its center as

$$m^k = \frac{1}{|C^k|} \sum_{x \in C^k} x.$$

5 For each solution $x \in C^k$ $(k = 1, 2, \cdots, K)$,

5.1 Generate trial solution u_1, u_2, and u_3 by using the parents from C^k.

5.2 Set $y = \arg \min_{u \in \{u_1, u_2, u_3\}} f(u)$.

5.3 Replace x by y if $f(y) < f(x)$.

6 If the stop condition is not satisfied, set $g = g + 1$ go to **Step 2**; otherwise, terminate and rerun the best solution found so far.

We would make the following comments to the above algorithm.

- In **Step 2**, the clustering process is re-initialized every G generations. The purpose is to prevent the clustering process tracking in local optima.
- In **Step 5.1**, CoDE generates three candidate solutions for each solution x by

$$u_{1,j} = \begin{cases} x_{r1,j} + F \cdot (x_{r2,j} - x_{r3,j}) & \text{if } rand < C_r \text{ or } j = j_{rnd} \\ x_j & \text{otherwise} \end{cases}$$

$$u_{2,j} = \begin{cases} x_{r1,j} + F \cdot (x_{r2,j} - x_{r3,j}) + F \cdot (x_{r4,j} - x_{r5,j}) & \text{if } rand < C_r \text{ or } j = j_{rnd} \\ x_j & \text{otherwise} \end{cases}$$

$$u_{3,j} = x_j + rand \cdot (x_{r1,j} - x_j) + F \cdot (x_{r2,j} - x_{r3,j})$$

where $j = 1, 2, \cdots, n$, j_{rnd} is a random index between 1 and n, $rand$ returns a random number in $[0.0, 1.0]$, $x_{r1} - x_{r5}$ are randomly selected parents from the same cluster as x, and F and C_r are two control parameters which are

randomly selected from $[F = 1.0, C_r = 0.1]$, $[F = 1.0, C_r = 0.9]$, and $[F = 0.8, C_r = 0.2]$.

- In **Step 6**, CoDE terminates when the function evaluation exceeds a given threshold.
- It is required that the minimum number of solutions in each cluster is 5 in the reproduction. If the number of solutions of a cluster is less than 5, the parents are selected from the whole population.

3. **Comparison with other mating strategies.** In this section, we compare the proposed CMS strategy with the following related strategies. **Random mating strategy (RND):** In this strategy, the parent solutions are randomly chosen from the whole population. The original CoDE algorithm actually utilizes this strategy. **Nearest neighbor strategy (NNS):** For a solution x, this strategy selects the closest $\lfloor \frac{N}{K} \rfloor$ solutions to form a mating pool for x, and the parents are randomly choose from the mating pool, where N is the number of population size and K is the number of niche the population can be divided. **Batch clustering based strategy (BCS):** As the CMS strategy, this strategy also uses the k-means method to partition the population. The difference is that the whole clustering process is implemented in the beginning of each iteration. All the strategies are incorporated into CoDE algorithm as CMS-CoDE does.

To access the performance of the compared strategies, the first 20 instances from the CEC 2005 test suite [17] are used for the comparison study. The parameters in the experiments are as follows: the dimension of the instances is $n = 30$ for all the 20 problems, all the algorithms stop after 30 independent runs with a maximum of $300,000$ function evaluations (FES), the population size is $N = 100$ for all algorithms, the number of clusters is $K = 3$ in the k-means clustering, and the k-means restarts every $G = 10$ generations. To have a fair comparison, the Wilcoxon's rank sum test at a 0.05 significance level is conducted, and $-$, $+$, and $\approx$ in the tables indicate that the performance of the corresponding method is better than, worse than, and similar to that of CMS, respectively. All the algorithms are executed in the workstation.

3.1. **Experimental results.** The experimental results are given in Table 1, and the population partitions of a typical run for BCS and CMS strategies with CoDE are plotted in Fig. 1 on two instances.

RND vs. CMS: From Table 1, we can see that CMS-CoDE performs better than RND-CoDE on 15 test instances and worse than RND-CoDE on 3 test instances. This suggests that, since CMS restricts the mating parents to be selected from the similar individuals, the mating strategy can improve the algorithm performance significantly.

NNS vs. CMS: It is clear from Table 1 that, CMS-CoDE outperforms NNS-CoDE on 7 instances and is outperformed by NNS-CoDE on 10 instances. In NNS, the parents are the closest ones with similar characteristics around the solution. Thus it may help to converge to optima quickly especially when there is no variable dependency in the problems. In CMS, the parents are likely to be the closest ones but there is still some probability that the parents are far away from each. This may help to keep population diversity in a sense. And this might be the reason to explain the different performances between NNS and CMS. Although the results are comparable, we can see from the next section that CMS-CoDE has a lower theoretical computational complexity than NNS-CoDE.

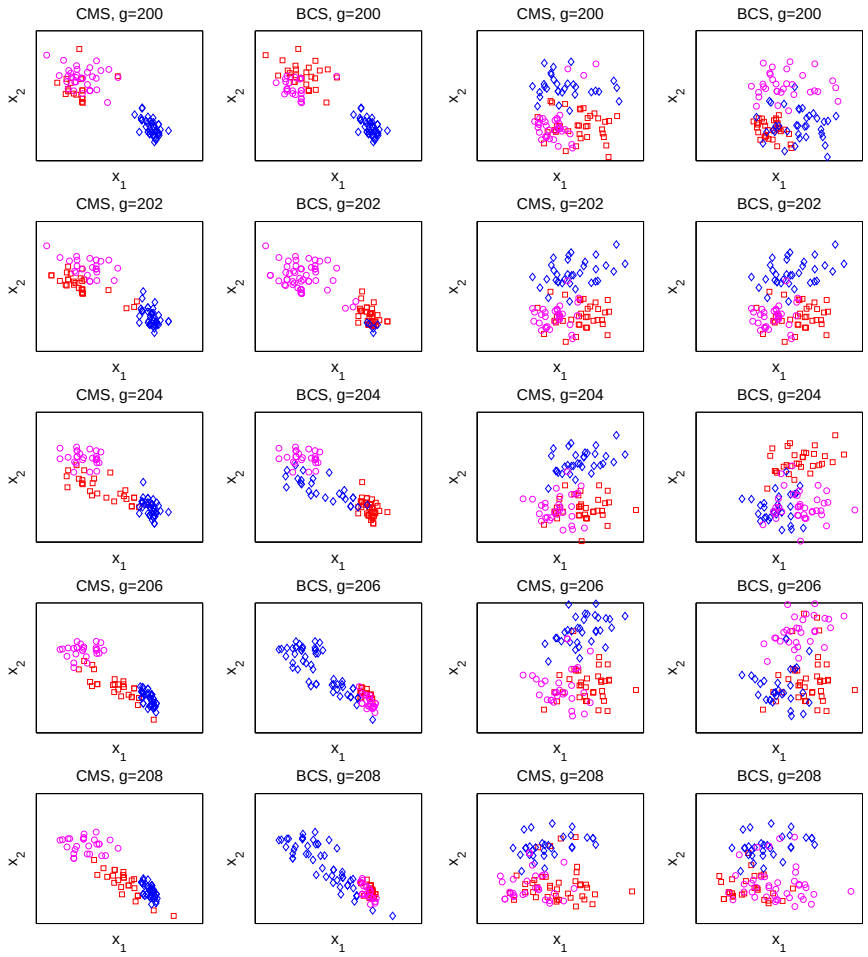
TABLE 1. The mean results of the compared methods over 30 independent runs on 20 test instances of 30 variables with 3000,000 FES.

	RND	NNS	BCS	CMS
F1	3.21e-08+	0.00e+00≈	0.00e+00≈	0.00e+00
F2	3.14e-01+	8.17e-04+	2.71e-05−	5.66e-05
F3	1.21e+05−	2.13e+05−	2.22e+05≈	2.64e+05
F4	1.08e+01+	3.74e+00+	1.87e-01−	2.72e-01
F5	3.90e+02−	1.00e+03≈	6.85e+02−	8.69e+02
F6	2.65e+01+	7.19e+01+	4.55e+01≈	3.80e+01
F7	4.70e+03+	4.70e+03+	4.70e+03≈	4.70e+03
F8	2.09e+01+	2.08e+01+	2.02e+01−	2.03e+01
F9	1.67e+01+	5.15e-06−	4.65e+00−	7.31e+00
F10	1.63e+02+	3.52e+01−	4.64e+01+	4.11e+01
F11	3.37e+01+	9.85e+00−	1.33e+01≈	1.35e+01
F12	1.88e+05+	1.54e+05+	7.16e+04≈	7.90e+04
F13	8.18e+00+	2.65e+00−	2.58e+00−	2.91e+00
F14	1.33e+01+	1.21e+01−	1.24e+01≈	1.23e+01
F15	6.40e+02+	5.14e+02+	4.71e+02≈	4.78e+02
F16	4.25e+02+	3.00e+02−	3.10e+02≈	3.15e+02
F17	4.66e+02+	3.03e+02−	3.16e+02≈	3.19e+02
F18	9.26e+02−	9.26e+02≈	9.24e+02≈	9.24e+02
F19	9.26e+02≈	9.26e+02−	9.25e+02≈	9.26e+02
F20	9.26e+02≈	9.24e+02−	9.25e+02≈	9.25e+02
+	15	7	1	
−	3	10	6	
≈	2	3	13	

BCS vs. CMS: It is surprising that BCS-CoDE performs slightly better than CMS-CoDE. Table 1 shows that there is not much difference between the results obtained by the two algorithms on 13 out of 20 test instances. The reason might be that the clustering results of k-means highly depend on the initial cluster centers and k-means is very likely to converge to local optima. Therefore, the mis-clustering in k-means leads some randomness to the population and prevent the premature of the population. Although BCS-CoDE is slightly superior to CMS-CoDE, it has a higher computational complexity according to the analysis in the next section.

With respect to the population partitions for BCS and CMS strategies with CoDE on F2 and F3, it is easy to find from Fig. 1 that, for CMS-CoDE, during the continuous 9 generations, the population partitions change a little; but for BCS-CoDE, it presents quite different partition models at each generation. The reason might be that the clustering operation of CMS-CoDE only iterates one time but that of BCS-CoDE iterate for many times. Actually, the stable population should be more helpful to generate the solutions with high quality.

3.2. **Time complexity.** The additional time complexity brought by the mating strategies is a major concern. The time complexity for the four strategies are as follows. *RND:* $O(N)$. *NNS:* For all solutions, the time complexity to calculate the distance between each pair is $O(N^2 \cdot n)$. For each solution, it choose the closes



(a) $F2$ (b) $F3$

FIGURE 1. Population partition of a typical run for BCS and CMS strategies with CoDE on (a) F2 and (b) F3.

$\lfloor \frac{N}{K} \rfloor$ solutions, and the time complexity is $O(N \cdot N \cdot \frac{N}{K} = \frac{1}{K}N^3)$. Therefore, the total time complexity is $O(N^2 \cdot n + \frac{1}{K}N^3)$. *BCS:* In k-means assignment step, the time complexity to assign each point to a cluster is $O(N \cdot K \cdot n)$. In the update step, the time complexity is $O((|C^1| + |C^2| + \cdots + |C^K|) \cdot n) = O(N\dot{n})$. Each solution will randomly select at most 5 parents from corresponding cluster and its time complexity is $O(N)$. Suppose the training steps in k-means is L. The total time complexity is $O((K+1) \cdot N \cdot n \cdot L + N)$. *CMS:* From the above analysis, we can see that the time complexity is $O((K+1) \cdot n \cdot N + N)$.

It is reasonable to assume that $K \ll N$ and $N \ll L$. The time complexities of NNS and BCS are much higher than those of RND and CMS. We can also see that although the time complexity of CMS is higher than that of RND, it is still linear according to N.

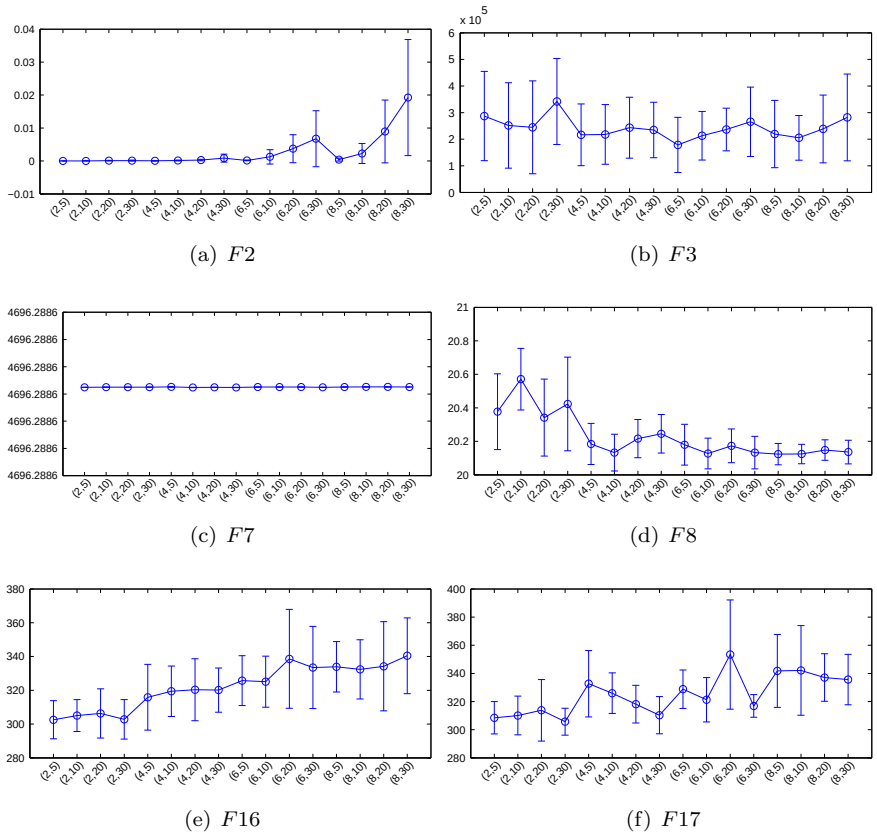
TABLE 2. The average CPU time (seconds) used by the four algorithms on F1-F20 with 300,000 function evaluations over 30 runs.

	RND	NNS	BCS	CMS
F1	11.68	12.84	17.61	12.99
F2	12.21	12.70	15.02	12.82
F3	12.93	12.83	15.11	13.09
F4	12.99	13.29	15.39	13.31
F5	16.07	15.18	17.08	15.09
F6	11.72	12.76	500.28	12.71
F7	14.54	15.73	17.33	15.58
F8	16.89	17.65	16.89	14.39
F9	14.98	14.46	126.06	12.82
F10	15.06	13.14	13.98	13.09
F11	61.29	58.76	58.64	57.58
F12	46.49	47.59	44.44	42.98
F13	13.06	13.15	14.53	13.00
F14	17.08	16.40	16.32	14.39
F15	113.89	115.58	180.81	115.32
F16	119.73	116.78	171.78	116.16
F17	120.10	117.87	452.28	117.31
F18	123.71	121.59	121.74	120.65
F19	149.03	152.96	152.79	147.89
F20	220.64	217.06	218.65	215.53

We also record the CPU run time in Table 2 although it depends on the algorithm implementation. It clearly shows that the additional cost consumed by CMS is not much by comparing RND and CMS strategies. On some instances, the CPU time of CMS is slightly less than that of RND. On all the instances, BCS needs more time than CMS which is consistent with the above analysis.

3.3. **Influence of control parameters.** There are two control parameters in CMS: the number of clusters K, and the number of generations G to restart the clustering process. This section studies the influence of the two parameters. Two unimodal functions F2 and F3, two multimodal functions F7 and F8, and two hybrid composition functions F16 and F17 are used to assess the performance. The population size is $N = 100$, the cluster number is set to $K = 2, 4, 6$ or 8, and the generation number to restart clustering is set to $G = 5, 10, 20$, or 30. The other parameters are the same as in the previous section.

Fig. 2 plots the error bars of the results obtained by CMS-CoDE with different combinations of control parameters over 30 runs on the 6 instances. On F2, it clearly shows that as K and G increase, the performance decreases. The reason is that F2 is unimodal problem and the best cluster number is 1, and k-means fails to capture the population structure with the given control parameters. On the contrary on F8, the performance increases as K and G increase. The reason is that F8 is a multimodal problem and large number of clusters may lead to better population partition. On F7, CMS-CoDE obtains very stable results and this indicates that CMS is not sensitive to the control parameters on the two problems. On F3, F16, and F17, the performance curves are not stable and the standard deviations are big



(a) $F2$ (b) $F3$

(c) $F7$ (d) $F8$

(e) $F16$ (f) $F17$

FIGURE 2. The error bars of the results obtained by CMS-CoDE with different combinations of control parameters (K, G) over 30 runs on some test instances.

on several combinations. We can also see from Fig. 2 that the performance is more sensitive to K than G. A moderate number of clusters is suitable.

4. **Conclusions.** In this paper, we proposed a strategy to integrate *statistical and machine learning (SML)* techniques to guide the search of *evolutionary algorithms (EAs)* efficiently. The idea is to combine the SML iteration and EA iteration together. The learning process and optimization process are performed alternatively. As an example, a general *clustering based mate selection (CMS)* assisted EA framework was proposed. In CMS, the population is partitioned into classes and the parents in the same class are allowed to do offspring reproduction. More specifically, a CMS utilizing k-means clustering technique was designed and integrated into a state-of-the-art EA. The experimental results suggested that CMS can improve the performance of existing EAs. The time complexity analysis also showed that the proposed approach does not bring much additional cost to the EA to improve.

It should be noted the current work is very preliminary and there are a variety of directions worth exploring. The combination of CMS and EAs should be improved. How to organize data for model building should be studied. Furthermore, it is worth to apply CMS strategy to multi-objective optimization problems.

REFERENCES

[1] T. Back, D. B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation*, Oxford University Press, 1997.

[2] K. Deb and D. E. Goldberg, An investigation of niche and species formation in genetic function optimization, in *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1989, 42–50.

[3] L. J. Eshelman and J. D. Schaffer, Preventing premature convergence in genetic algorithms by preventing incest, in *International Conference on Genetic Algorithms*, 1991, 115–122.

[4] C. M. Fernandes and A. C. Rosa, Evolutionary algorithms with dissortative mating on static and dynamic environments, *Advances in Evolutionary Algorithms*, 2008, 181–206.

[5] S. F. Galán, O. J. Mengshoel and R. Pinter, A novel mating approach for genetic algorithms, *Evolutionary Computation*, **21** (2013), 197–229.

[6] A. Gog, C. Chira, D. Dumitrescu and D. Zaharie, Analysis of some mating and collaboration strategies in evolutionary algorithms, in *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, 2008, 538–542.

[7] K. S. Goh, A. Lim and B. Rodrigues, Sexual selection for genetic algorithms, *Artificial Intelligence Review*, **19** (2003), 123–152.

[8] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second edition. Springer Series in Statistics. Springer, New York, 2009.

[9] Y. Jin, Surrogate-assisted evolutionary computation: Recent advances and future challenges, *Swarm and Evolutionary Computation*, **1** (2011), 61–70.

[10] P. Larranaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2002.

[11] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Ed. University of California Press, (1967), 281–297.

[12] G. Ochoa, C. Mädler-Kron, R. Rodriguez and K. Jaffe, Assortative mating in genetic algorithms for dynamic problems, in *Applications of Evolutionary Computing*, Springer, 2005, 617–622.

[13] T. S. Quirino, *Improving Search in Genetic Algorithms Through Instinct-Based Mating Strategies*, Ph.D. dissertation, The University of Miami, 2012.

[14] T. Quirino, M. Kubat and N. J. Bryan, Instinct-based mating in genetic algorithms applied to the tuning of 1-nn classifiers, *IEEE Transactions on Knowledge and Data Engineering*, **22** (2010), 1724–1737.

[15] J. Sanchez-Velazco and J. A. Bullinaria, Sexual selection with competitive/co-operative operators for genetic algorithms, in *Neural Networks and Computational Intelligence(NCI)*. ACTA Press, 2003, 191–196.

[16] R. Sivaraj and T. Ravichandran, A review of selection methods in genetic algorithm, *International Journal of Engineering Science and Technology (IJEST)*, **3** (2011), 3792–3797.

[17] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger and S. Tiwari, *Problem Definitions and Evaluation Criteria for the cec 2005 Special Session on Real-Parameter Optimization*, Tech. rep., Nanyang Technological University, Singapore and Kanpur Genetic Algorithms 369 Laboratory, IIT Kanpur, 2005.

[18] C.-K. Ting, S.-T. Li and C. Lee, On the harmonious mating strategy through tabu search, *Information Sciences*, **156** (2003), 189–214.

[19] S. Wagner and M. Affenzeller, Sexualga: Gender-specific selection for genetic algorithms, in *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, **4** (2005), 76–81.

[20] R. Wang, P. J. Fleming and R. C.Purshousea, General framework for localised multi-objective evolutionary algorithms, *Information Sciences*, **258** (2014), 29–53.

[21] Y. Wang, Z. Cai and Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, *IEEE Transactions on Evolutionary Computation*, **15** (2011), 55–66.

E-mail address: jyzhang@stu.ecnu.edu.cn
E-mail address: amzhou@cs.ecnu.edu.cn
E-mail address: gxzhang@cs.ecnu.edu.cn
E-mail address: jxzhanghu@126.com