A Self-organizing Multi-objective Particle Swarm Optimization Algorithm for Multimodal Multi-objective Problems

Jing Liang[1], Qianqian Guo[1], Caitong Yue[1], Boyang Qu[2,*], Kunjie Yu[1]

[1] School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China, email: liangjing@zzu.edu.cn; 18137780112@163.com; zzuyuecaitong@163.com; yukunjie@zzu.edu.cn.

[2] School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China, email: qby1984@hotmail.com.

Abstract. To solve the multimodal multi-objective optimization problems which may have two or more Pareto-optimal solutions with the same fitness value, a new multi-objective particle swarm optimizer with a self-organizing mechanism (SMPSO-MM) is proposed in this paper. First, the self-organizing map network is used to find the distribution structure of the population and build the neighborhood in the decision space. Second, the leaders are selected from the corresponding neighborhood. Meanwhile, the elite learning strategy is adopted to avoid premature convergence. Third, a non-dominated-sort method with special crowding distance is adopted to update the external archive. With the help of self-organizing mechanism, the solutions which are similar to each other can be mapped into the same neighborhood. In addition, the special crowding distance enables the algorithm to maintain multiple solutions in the decision space which may be very close in the objective space. SMPSO-MM is compared with other four multi-objective optimization algorithms. The experimental results show that the proposed algorithm is superior to the other four algorithms.

Keywords: Self-organizing, Multimodal multi-objective problems, Multi-objective particle swarm optimizer, Elite learning strategy.

1 Introduction

In real world, many problems have two or more conflicting objectives to be optimized. For these problems, an improvement of one objective may lead to degradation in others. Multi-objective optimization algorithms provide a best tradeoff solution set instead of a single solution. The solution set is known as the Pareto solution set (PS) in the search space and the set of all the vectors in the objective space corresponding to the PS is called Pareto front (PF).

---

* Corresponding author

There are some problems which have more than one solution corresponding to the same point in the objective space. The literature [1] identified the existence of problems with multiple PSs. Liang referred to this class of problems as multimodal multi-objective problems [2]. Niching methods are applied to locate the multiple optimal solutions and keep them from being deleted. However, classic niching methods such as fitness sharing, crowding, and speciation are sensitive to the value of niching parameters [3]. Yue adopted an index-based ring topology [4] to form stable niches without any niching parameters and proposed non-dominated-sort method with special crowding distance (Non-dominated-scd-sort) to maintain multiple solutions [5].

For most multi-objective problems, under mild conditions, both the PS and the PF are (m−1) dimensional piecewise continuous manifolds [6]. Based on the regularity, many multi-objective evolutionary algorithms have been proposed. MOEA/D decomposed a multi-objective problem into a set of subproblems by a set of predefined weight vectors [7]. RM-MEDA clustered the PS by using the local principal component analysis and sampled new solutions from the built model [8]. Zhou used self-organizing map (SOM) to establish the neighborhood relationship and to generate offspring with the neighboring solutions [9].

Particle swarm optimization algorithm is a simple and robust algorithm [10]. The fast convergence of particle swarm optimization leads to the loss of diversity to some extent. Some mechanisms have been proposed to address this issue. Decomposition strategy maintains diversity by ensuring each sub-region has a solution in the objective space [11-13]. Mutation operator helps the algorithm to jump out of local optimal location [14-15]. Different neighbor relationships of particles have been introduced to select neighbor best and sharing information [16-17].

When solving multimodal multi-objective problems, the neighbor leaders play an important role in local search. Inspired by the characteristics of multi-objective optimization problems and the neighbor property of SOM, a self-organizing multi-objective particle swarm optimization algorithm for solving multimodal multi-objective problems (SMPSO-MM) is proposed in this paper. Self-organizing map network is used to gather good and similar solutions together in hidden layers in MOPSO and to build neighboring relationship in decision space.

The rest of this paper is organized as follows: Section 2 briefly describes the multi-objective problems and multi-objective particle swarm optimization algorithm. Section 3 presents the self-organizing multi-objective particle swarm algorithm in detail. Experimental results and analysis are shown in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Works

2.1 Multi-objective Problems(MOPs)

A continuous multi-objective optimization problem can be formulated as follows:

$$\text{Min } \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})) \text{ s.t. } \boldsymbol{x} = (x_1, x_2, ..., x_n) \in \Omega \quad (1)$$

where $\Omega=[a_i, b_i]^n$ is feasible region of the search space; $\boldsymbol{x}$ is a n-dimensional decision vector bounded in Ω; m is the number of objective functions. F: $\Omega \rightarrow R^m$ consists of m objective functions to be optimized and R^m denotes the objective space. In the following texts, some important definitions for multi-objective problems are given.

Definition1. (Pareto dominance): A decision vector $\boldsymbol{x}$ is said to dominate decision vector $\boldsymbol{y}$ (denoted as $\boldsymbol{x} \prec \boldsymbol{y}$). If and only if

$$(\forall i \in \{1,2,...,m\}: f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})) \wedge (\exists j \in \{1,2,...,m\}: f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})) \tag{2}$$

Definition2. (Pareto optimal solution): A decision vector $\boldsymbol{x}$ is said to be Pareto optimal solution with respect to Ω if

$$\neg\exists \boldsymbol{y} \in \Omega : \boldsymbol{y} \prec \boldsymbol{x} \tag{3}$$

Definition3. (Pareto optimal set, **PS**): is defined as:

$$\mathbf{PS} = \{\boldsymbol{x} \in \Omega | \neg\exists \boldsymbol{y} \in \Omega : \boldsymbol{y} \prec \boldsymbol{x}\} \tag{4}$$

Definition4. (Pareto front, **PF**) is defined as:

$$\mathbf{PF} = \{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbf{PS}\} \tag{5}$$

2.2 Multi-objective Particle Swarm Optimization Algorithm(MOPSO)

In MOPSO, each particle represents a potential solution. Assuming there are N particles in the swarm and the searching space is n-dimensional hyperspace. The position and velocity of particle i are represented as $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$ and $\boldsymbol{v}_i = (v_{i1}, v_{i2}, ..., v_{in})$ respectively. They are updated according to the following equations:

$$\boldsymbol{v}_i(t+1) = w^* \boldsymbol{v}_i(t) + c_1 r_1 (\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i(t)) + c_2 r_2 (\boldsymbol{x}_{nbest_i} - \boldsymbol{x}_i(t)) \tag{6}$$

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \tag{7}$$

where t is the iteration; w is inertia factor; c_1 and c_2 are two constants which affect acceleration; r_1 and r_2 are two random variables in the range (0,1); *pbest* and *nbest* represent personal best and neighbor best position of the i^{th} particle, respectively.

3 The Details of SMPSO-MM

3.1 Leaders Selection in Neighborhood

The proposed SMPSO-MM algorithm uses SOM to find the distribution of current population (**POP**) and external archive (**EXA**). Newly generated non-dominated solutions are training sets (**TS**) for updating SOM model. SOM clusters similar solutions into same neighborhood. Non-dominated-scd-sort method [5] is used to selected *nbest* from neighborhood. **Algorithm 1** presents the details of selecting *nbest* .

The process of composing *nbest* pool is illustrated in Fig.1. The solutions assigned to neighboring neurons of winning neuron *u* compose the *nbest* pool. The advantages of this selection method are presented as: 1) the weights are constantly updated by new non-dominated solutions, so SOM can reflect the distribution of PS more accurately; 2) the *nbest* is close to particle in the decision space; 3) different leaders can promote the diversity and locate more multimodal solutions.

Algorithm 1: The procedure of selecting *nbest*

1. **Find the index of neighboring neurons**

   Calculate the geographical distance $\left\| z^{u} - z^{u'} \right\|_2$, I_k^u is the index of the k^{th} nearest neuron to neuron *u* in the representation layer.

2. **Update SOM model**

   2.1. Update learning rate $\eta = \eta_0 * (1 - \frac{t}{maxT})$, learning radius $\sigma = \sigma_0 * (1 - \frac{t}{maxT})$ / t is the current iteration, *maxT* is maximum iteration.

   2.2. Find the winning neuron of $\boldsymbol{x} \in \mathbf{TS}$: $u' = \arg \min\limits_{1 \le u \le N} \left\| \boldsymbol{x} - \boldsymbol{w}^{u} \right\|_2$

   2.3. Locate the neighborhood neurons to be updated: $\mathrm{U} = \{1 \le u \le N \wedge \left\| z^{u} - z^{u'} \right\|_2 < \sigma\}$

   2.4. Update the weights in U. $\boldsymbol{w}_{t+1}^{u} = \boldsymbol{w}_t^{u} + \eta(t) * \exp(\left\| z^{u} - z^{u'} \right\|_2)(\boldsymbol{x} - \boldsymbol{w}_t^{u})$

3. **POP and EXA partition**

   3.1. Map $\boldsymbol{x}_i \in \mathbf{POP}$ to SOM and find the winning neuron *u*. Each unit is assigned to one particle;

   3.2. Map $\boldsymbol{x}_i \in \mathbf{EXA}$ to SOM and *unit2nbest* records the index of particles assigned to each neuron. The number of particles assigned to each unit is unlimited.

4. **Select *nbest* for particles**

   4.1. nPool= unit2nbest{ I_k^u };/ nPool records the *nbest* pool of particle *i*;

   4.2. Select the first one as *nbest* according to Non-dominated-scd-sort method.

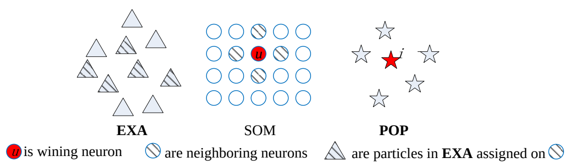


Fig.1. The process of composing *nbest* pool

3.2 The Elite Learning Strategy

The elite learning strategy is applied to offspring generation. Operating mutation on the best location can enhance global search ability and help the algorithm to jump out of local optimal location.

$$x_j = nbest_j + Gauss(0, pr^2) \times (b_j - a_j) \quad \text{if} \quad rand < pr \tag{8}$$

where a_j and b_j are the upper and lower bounds of j^{th} dimension. The $Gauss(0, pr^2)$ is a random number of a Gaussian distribution with a zero mean and a stand deviation *pr.* The *pr* decreases linearly with the iteration increases. According to the Gaussian distribution, the greater value of *pr* can make the position have a larger variation range, which is beneficial for global search in the early stage. In the later stage of evolution, variation range is small which is beneficial for local search.

3.3 Procedure of SMPSO-MM

The procedure of SMPSO-MM is described in **Algorithm 2**. The weights of SOM are initialized as **POP**. The *nbest* selected from the neighborhood leads the particles flying to promising locations. The elite learning strategy is applied in generating offspring operator to increase the diversity of population. And Non-dominated-scd-sort method [5] is adopted to update **EXA** and keep diversity both in decision and objective space. The Non-dominated-scd-sort method firstly sorts the particles according to dominance relationship. Then the crowding distance in decision space and objective space are calculated. The particles with larger crowding distance are preferred. Details of the method can be referred to the literature [5].

Algorithm 2: The procedure of SMPSO-MM

1. **Initialize population and self-organizing map**

1.1 Initialize $\mathbf{POP} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N\}$, $\mathbf{v} = \{\boldsymbol{v}^1, \boldsymbol{v}^2, ..., \boldsymbol{v}^N\}$; **EXA**=**POP**; **Pbest=POP**.

1.2 Initialize SOM model: $\{\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^N\}$=**POP** ; **TS=POP**; learning rate η_0, learning radius σ_0.

2. **Optimization loop**

2.1. Select *nbest* according to **Algorithm 1**.

2.2. Generate offspring according to equation (6-7) and elite learning strategy.

2.3. Update the **EXA** and **TS**

$\mathbf{tmpEXA} = \mathbf{EXA}(t) \cup \mathbf{POP}(t+1)$; /*t* is the current iteration

EXA(*t*+1)=non-dominated particles in **tmpEXA**;

TS=EXA(*t*+1)\ **EXA**(*t*);

3. **Stop if a stop criterion is satisfied**, **otherwise repeat the Optimization loop**.

4 Experimental Results

The multimodal multi-objective test functions used in this paper include MMF1-MMF8 [5] and three other functions named SYM-PART simple [18], SYM-PART rotated [18] and Omni-test function [19] with *n*=3.

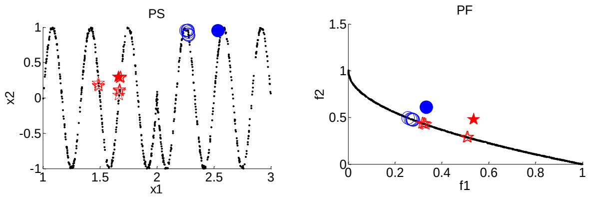
The Pareto set proximity (*PSP*) [5] and inverted generational distance in the objective space (*IGDf*) are adopted as metrics to evaluate the performance of the algorithms. The big value of *PSP* means the convergence and diversity of obtained solutions in decision space are good. The small value of *IGDf* means the obtained solutions in objective space are well distributed.

SMPSO-MM is compared with other four algorithms including MOPSO/D [11], Omni-optimizer [19], DN-NSGAII [2] and MO-Ring-PSO-SCD [5]. The parameters of algorithms are the same to the corresponding literature.

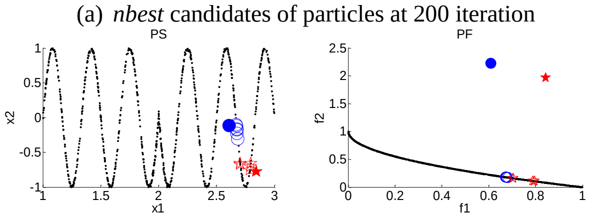
For SMPSO-MM, the parameters of PSO are same as literature [5] : $c_1 = c_2 = 2.05$, $w = 0.7298$; the parameters of SOM are same as literature [9] : topological structure 1×100 , $\eta_0 = 0.7$, $\sigma_0 = \mathrm{sqrt}(1^2 + 100^2)/2$; the value of *pr* in elite learning decreases linearly from 0.2 to 0.05 with the iteration increases, the value of *pr* is set according to the experimental results. For all algorithms, the population size *N* is 100; the EXA size is 800; Maximum evaluation number is 60000. All experiments are carried out 25 times independently.

4.1 The Rationality of Neighborhood Built by SOM

To verify the rationality of the neighborhood relationship, the distribution of *nbest* candidates in the evolutionary process for MMF2 are shown in Fig.2. The figure shows that the *nbest* candidates selected based on the neighboring relationship are close to particle both in the decision space and objective space. It verifies the rationality of the neighborhood relationship as the particle can fly to the near position by learning from neighboring leader to promote local search.



(a) *nbest* candidates of particles at 200 iteration



(b) *nbest* candidates of particles at 400 iteration

. is solution in **EXA**; ★ is particle *i*, ☆ is *nbest* candidate of particle *i*;
● is particle *j*, ○ is *nbest* candidate of particle *j;*

Fig.2. *nbest* candidates of particles in the process of evolution

4.2 Results of Compared Algorithms

In Fig.3, MOPSO/D, Omni-optimizer, DN-NSGAII, MO-Ring-PSO-SCD and SMPSO-MM are numbered 1, 2, 3, 4 and 5 respectively. The results show that mean *PSP* values of SMPSO-MM are the highest for all test functions. MO-Ring-PSO-SCD ranks second. The *IGDf* and *rank* shown in Table 1 reveal that SMPSO-MM is the second best and Omni-optimizer obtains the best distribution in decision space. The *rank* value is the mean rank on all test functions. In fact, the performances of compared algorithms are close to each other except MOPSO/D. The reason is that the four algorithms all consider the distribution in the objective space.

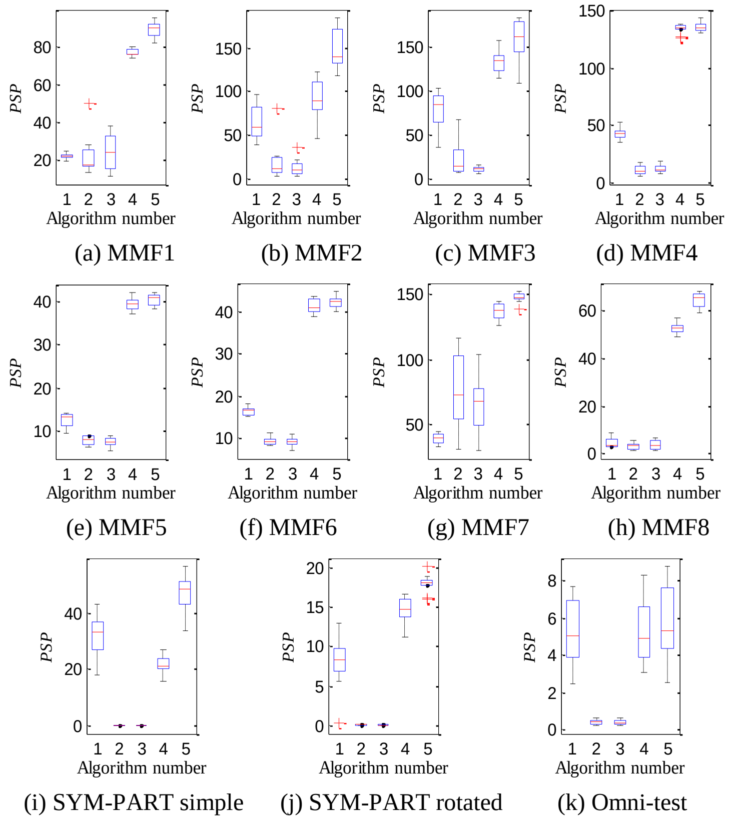


Fig.3. The box-plots of *PSP* values of different algorithms. The numbers on the horizontal axis of each plot indicate the following algorithms: 1=MOPSO/D, 2=Omni-optimizer, 3=DN-NSGAII, 4=MO-Ring-PSO-SCD, 5=SMPSO-MM.

4.3 The Effect of The Size of External Archive

Different external archive (EXA) sizes affect the performance of the algorithm especially for multimodal multi-objective problems. Fig.4 shows the result of *PSP* values with different external sizes. The test functions are MMF1, MMF4, MMF8 and SYM-PART simple. *PSP* values increase as the EXA size grows for most test functions. The large size of EXA allows more non-dominated solutions to locate more solutions. And the small size increases the challenge to maintain solutions uniformly.

In order to study how the performances of the algorithm are affected by the EXA size, different EXA sizes of all algorithms are applied on MMF3, MMF8, SYM-

PART simple and SYM-PART rotated test functions. The results reported in Fig.5 confirm that the performances of the algorithms are affected by varying EXA sizes. SMPSO-MM is superior to other algorithms for most sizes of EXA.

Table 1. The *IGDf* values of different algorithms

Functions	MOPSO/D	Omni-optimizer	DN-NSGAII	MO-Ring-PSO-SCD	SMPSO-MM
MMF1	2.30E-03[5] (2.77E-04)	6.69E-04[2] (3.08E-05)	8.46E-04[4] (5.73E-05)	7.95E-04[3] (4.25E-05)	**6.61E-04**[1] (2.98E-05)
MMF2	3.62E-03[4] (4.87E-04)	2.86E-03[2] (7.37E-03)	**1.84E-03**[1] (3.87E-03)	5.22E-03[5] (6.45E-04)	2.89E-03[3] (2.04E-04)
MMF3	3.38E-03[4] (2.05E-04)	**6.29E-04**[1] (2.00E-05)	1.04E-03[2] (8.43E-04)	4.04E-03[5] (5.10E-04)	2.33E-03[3] (1.19E-04)
MMF4	1.65E-03[5] (1.21E-04)	6.40E-04[2] (4.72E-05)	7.79E-04[4] (6.04E-05)	6.62E-04[3] (7.09E-05)	**6.29E-04**[1] (4.15E-05)
MMF5	2.34E-03[5] (2.58E-04)	6.51E-04[2] (1.43E-05)	8.03E-04[4] (3.93E-05)	7.36E-04[3] (1.63E-05)	**6.50E-04**[1] (2.64E-05)
MMF6	2.00E-03[5] (1.22E-04)	6.44E-04[2] (2.24E-05)	7.84E-04[4] (3.58E-05)	6.81E-04[3] (2.04E-05)	**6.27E-04**[1] (2.88E-05)
MMF7	2.06E-03[5] (2.52E-04)	6.69E-04[2] (2.01E-05)	9.19E-04[4] (4.19E-05)	6.71E-04[3] (2.46E-05)	**6.44E-04**[1] (2.14E-05)
MMF8	1.46E-03[4] (1.07E-04)	**7.23E-04**[1] (2.28E-05)	8.47E-04[3] (3.18E-05)	1.16E-03[5] (6.36E-05)	8.42E-04[2] (1.91E-05)
SYM-PART simple	7.55E-03[4] (1.76E-03)	**2.89E-03**[1] (2.93E-04)	2.94E-03[2] (3.22E-04)	9.15E-03[5] (1.40E-03)	3.96E-03[3] (5.85E-04)
SYM-PART rotated	1.67E-02[5] (3.52E-03)	**3.12E-03**[1] (3.88E-04)	3.67E-03[2] (3.12E-04)	1.39E-02[4] (2.60E-03)	1.15E-02[3] (1.56E-03)
Omni-test	9.98E-01[5] (2.28E-04)	9.97E-01[3] (2.32E-04)	9.97E-01[4] (3.40E-04)	**9.79E-01**[1] (1.81E-03)	9.82E-01[2] (1.69E-03)
rank	4.64	1.73	3.09	3.63	1.91

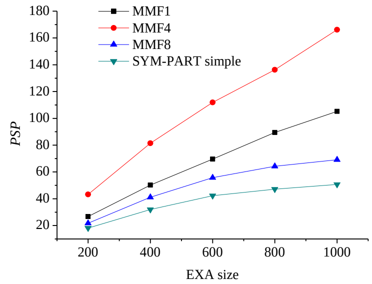
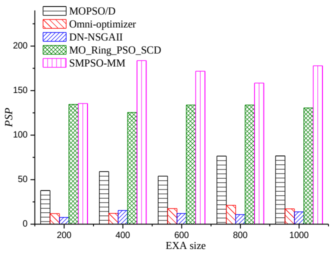
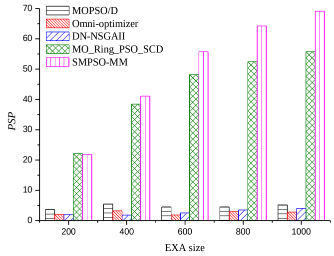


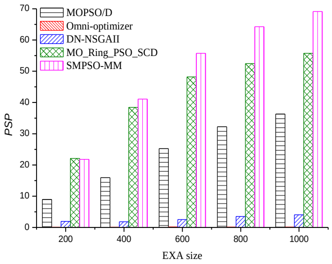
Fig.4. The *PSP* values with different external sizes of SMPSO-MM



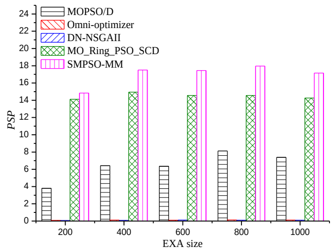
(a) *PSP* value on MMF3



(b) *PSP* value on MMF8



(c) *PSP* value on SYM-PART simple



(d) *PSP* value on SYM-PART rotated

Fig.5. The *PSP* values with different external size of different algorithms.

5 Conclusion

In this paper, we proposed a new multi-objective particle swarm algorithm with self-organizing mechanism to solve multimodal multi-objective problems. A new strategy to build the neighborhood and select neighborhood leaders is applied in MOPSO. The SOM network can reflect the distributions of current particles and non-dominated solutions in the decision space. The elite learning strategy promotes the diversity and avoids the premature of algorithm. Experimental results show that the proposed algorithm can locate the multiple solutions in the decision space and have a good distribution on Pareto front in the objective space for multimodal multi-objective problems.

Acknowledgments. The work is supported by National Natural Science Foundation of China (61473266 and 61673404), Project supported by the Research Award Fund for Outstanding Young Teachers in Henan Provincial Institutions of Higher Education of China (2014GGJS-004) and Program for Science & Technology Innovation Talents in Universities of Henan Province in China (16HASTIT041 and 16HASTIT033), Scientific and Technological Project of Henan Province (152102210153).

References

1. Preuss M., Kausch C., Bouvy C., Henrich F.: Decision space diversity can be essential for solving multiobjective real-world problems. Springer Berlin Heidelberg, (2010).
2. Liang J. J., Yue C. T., Qu B. Y.: Multimodal multi-objective optimization: A preliminary study. In: IEEE Congress on Evolutionary Computation. 2451-2461(2016).
3. Li X., Epitropakis M. G., Deb K., Engelbrecht A.: Seeking multiple solutions: An updated survey on niching methods and their applications. IEEE Transactions on Evolutionary Computation, 21(4), 518-538(2017).
4. Li X.: Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Transactions on Evolutionary Computation, 14(1):150-169(2010).
5. Yue C. T., Liang J. J., Qu B. Y.: A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems, IEEE Transaction on Evolutionary Computation. (DOI: 10.1109/TEVC.2017.2754271) (2017).
6. Li H., Zhang Q. F.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 13(2), 284-302. (2009).
7. Wang L. P., Zhang Q. F., Zhou A. M., Gong M. G., Jiao L. C.: Constrained subproblems in decomposition based multiobjective evolutionary algorithm. IEEE Transactions on Evolutionary Computation, 20(3), 475-480. (2016).
8. Zhang Q. F., Zhou A. M., Jin Y. C.: RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation, 12(1), 41-63(2008).
9. Zhang H., Zhou, A. M., Song, S. M., Zhang, Q. F., Gao X. Z., Zhang, J.: A self-organizing multiobjective evolutionary algorithm. IEEE Transactions on Evolutionary Computation, 20(5), 792-806(2016).
10. Kennedy J., Eberhart R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, 4, 1942-1948(1995).
11. Dai C., Wang Y. P., Ye M.: A new multi-objective particle swarm optimization algorithm based on decomposition. Information Sciences, 325,541-557(2015).
12. Fei L. I., Liu J. C., Shi. H. T., Zi-ying F.U.: Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution. Control & Decision, 32(3), 403-410(2017).
13. Wei L. X., Fan R., Li X.: A novel multi-objective decomposition particle swarm optimization based on comprehensive learning strategy. In: 36th Chinese Control Conference. 2761-2766(2017).
14. Dong W. Y., Kang L. L., Zhang W. S.: Opposition-based particle swarm optimization with adaptive mutation strategy. Soft Computing, 21(17), 5081-5090(2017).
15. Chen C. C.: Optimization of zero-order TSK-type fuzzy system using enhanced particle swarm optimizer with dynamic mutation and special initialization. International Journal of Fuzzy Systems, (DOI: 10.1007/s40815-018-0453-z)(2018).
16. Liang J. J., Suganthan P. N.: Dynamic multi-swarm particle swarm optimizer with local search. In: IEEE Congress on Evolutionary Computation. 1,522-528(2005).
17. Zhao S. Z., Suganthan P. N.: Two-lbests based multi-objective particle swarm optimizer. Engineering Optimization, 43(1), 1-17(2011).
18. Rudolph G., Naujoks B., Preuss M.: Capabilities of EMOA to detect and preserve equivalent pareto subsets. Springer Berlin Heidelberg, (2007).
19. Deb K., Tiwari S.: Omni-optimizer: A procedure for single and multi-objective optimization. Lecture Notes in Computer Science, 3410, 47-61(2005).