MMOGA for Solving Multimodal Multiobjective Optimization Problems with Local Pareto Sets

C. T. Yue, J. J. Liang
School of Electrical Engineering
Zhengzhou University
Zhengzhou, China
zzuyuecaitong@163.com,
liangjing@zzu.edu.cn

P. N. Suganthan
School of Electrical Electronic Engineering
Nanyang Technological University
Singapore
epnsugan@ntu.edu.sg

B. Y. Qu
School of Electric and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China
qby1984@hotmail.com

K. J. Yu
School of Electrical Engineering
Zhengzhou University
Zhengzhou, China
yukunjie@zzu.edu.cn

S. Liu
Medical Equipment Department
Henan Provincial People's Hospital
283321428@qq.com

***Abstract*—Multiobjective optimization problems with multiple equivalent global Pareto solutions or with at least one local Pareto solution are called multimodal multiobjective optimization problems (MMOP). Most of the existing multimodal multiobjective algorithms can only find global Pareto solutions. However, the local Pareto solutions are of great significance when the global ones are impracticable. This paper proposes a Multimodal Multiobjective Genetic Algorithm (MMOGA) to find both global and local Pareto solutions. In MMOGA, only individuals in the same niche can mate and compete with each other, thus enabling the population to evolve in local areas. Experimental results show that the proposed algorithm can find both global and local Pareto sets of MMOPs.**

Keywords—Multimodal multiobjective optimization; local Pareto solution; genetic algorithm

I. INTRODUCTION

Multiobjective optimization problems are almost anywhere in our daily life, like path planning optimization [1], scheduling optimization [2], feature selection optimization [3] and so on. Generally, the objectives to be optimized are conflicting. In other words, improving one objective value will dedicate others. Therefore, it is impossible to find one solution achieving the best performance for every objective. Multiobjective optimization algorithms usually provide a trade-off solution set for decision-makers. The best trade-off solution set is called Pareto optimal Set (PS) and the mapping set of PS in objective space is called Pareto Front (PF).

Multimodal optimization problem usually refers to single objective problem which has multiple optima. These optima include global and local ones. Similarly, some multiobjective optimization problems have multiple PSs including global PS and local PS. These problems are called multimodal multiobjective optimization (MMO) problems.

There are some prior works on MMO. Deb constructed several types of test problems in Ref. [4] including MMO problems. However, he only analyzed the difficulties facing multiobjective genetic algorithms. No MMO algorithms were proposed in Ref. [4]. Subsequently, he proposed Omni-optimizer [5] which can deal with multiple kinds of problems including uni/multimodal single/multiobjective problems. Liang et al. [6] proposed DN-NSGAII specifically for solving MMOPs. It is a preliminary study on MMO, so the performance of DN-NSGAII is not very satisfactory. Then, Yue et al. [7] proposed MO_Ring_PSO_SCD for solving MMO problems. MO_Ring_PSO_SCD uses ring topology to avoid falling into a local area and special crowding distance to keep multiple equivalent Pareto optimization solutions. In addition, Yue et al. proposed eight MMO test problems and designed performance indicator for MMO. Subsequently, many MMO algorithms are proposed. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies named TriMOEA-TA&R [8] was proposed by Liu et al. In Ref. [8], several new MMOPs were also proposed. In addition, a double-niched evolutionary algorithm is proposed by Liu et al. [9]. Then, a decomposition-based evolutionary algorithm is proposed by Tanabe et al. [10]. Another MMODE algorithm is proposed by Liang et al. [11].

Although there is much research on MMO, they all focus on how to find multiple equivalent global PSs. Seldom pays attention to the MMO problems with local PS. In Ref. [12], Yue designed several scalable MMO test problems with local PSs. Liu proposed DNEA-L searching from local Pareto optimal solutions on polygon-based problems. However, the DNEA-L was not tested on other types of MMO test problems. In fact, the local PS is not easy to be found if nondominated sorting method is used in the whole population. It is because local Pareto optimal solutions are dominated by the global Pareto optimal ones and they are deleted in the environmental selection process. However, in many cases, local Pareto solutions are preferred by decision-makers when the global ones are infeasible or too expensive to be obtained. Therefore, it is meaningful to study MMO algorithm finding local PSs.

In this paper, an MMO genetic algorithm is proposed to find both global and local PSs. A niching method is used in the decision space so that the population can evolve in a local scope. Instead of sorting the individuals in the whole population, the individuals are only compared with their neighborhood. Therefore, the local Pareto optimal solution will not be deleted in environmental selection.

The rest of this paper is organized as follows. Section II introduces the related definitions of MMO. Section III analyses

The work is supported by National Natural Science Foundation of China (61876169, 61922072, 61976237 and 61806179) and Scholarship under the State Scholarship Fund by the China Scholarship Council. (Corresponding author: Jing Liang.)

the difficulties of solving MMOPs with local PSs. The details of the proposed algorithm MMOGA are introduced in Section IV. Section V presents the experimental results and effectiveness analysis. Finally, the conclusion and future works are given in Section VI.

II. RELATED DEFINITIONS

This section introduces the definitions related to MMO including the definitions of multiobjective optimization, multimodal multiobjective optimization, nondominated relationship, global and local PS and PF.

Multiobjective optimization problems refer to those with more than one objective to be optimized. Without loss of generality, a minimization multiobjective optimization problem with m objectives and n decision variables can be formulated as:

$$\begin{aligned} &\min \ \ \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ &\text{s.t.} \ \ \mathbf{x} \in S \end{aligned} \quad (1)$$

where $\mathbf{x} \in S \subset \mathbf{R}^n$ is n-dimensional decision variable vectors and S represents search space. $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$ are the m objectives to be optimized. Since there is more than one objective, it is inappropriate to compare different solutions according to one certain objective value. The most popular method to compare solutions in multiobjective optimization is nondominated sorting. Without a special explanation, the optimization problems studied in this paper are all minimization problems.

Definition 1 Dominate relationship: Given two feasible solutions $\mathbf{x}_1$ and $\mathbf{x}_2$, $\mathbf{x}_1$ dominate $\mathbf{x}_2$ if : (1) $\mathbf{x}_1$ is not worse than $\mathbf{x}_2$ for all objectives i.e. $f_i(\mathbf{x}_1) \le f_i(\mathbf{x}_2)$ for $i = 1, \ldots, m$; (2) $\mathbf{x}_1$ is better than $\mathbf{x}_2$ for at least one objective i.e. $\exists i \in [1, m]$, $f_i(\mathbf{x}_1) \le f_i(\mathbf{x}_2)$.

If a certain solution is not dominated by any other solutions, it is called nondominated solution. The set of all nondominated solutions is the best trade-off solution set i.e. PS and the set of images of PS in objective space is PF.

Definition 2 Local Pareto optimal set (Local PS): For a solution set L_{ps}, if its arbitrary member $\mathbf{x}_L$ is not dominated by any neighborhood solution $\mathbf{y}$ ($\|\mathbf{y} - \mathbf{x}\|_\infty \le \sigma$, σ is a very small positive value), the solution set L_{ps} is called local PS.

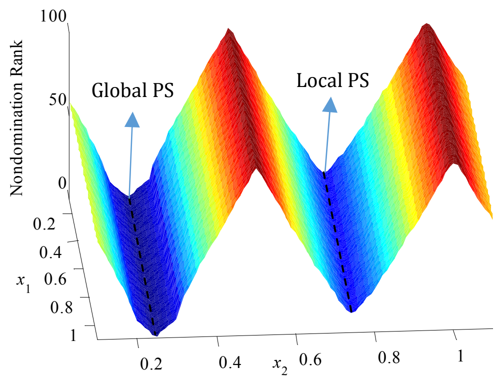
Definition 3 Global Pareto optimal set (Global PS): For a solution set G_{ps}, if its arbitrary member $\mathbf{x}_G$ is not dominated by any solution in the whole feasible space, the solution set G_{ps} is called global PS.

The image of local PS in objective space is called **Local PF** and the image of Global PS is called **Global PF**.

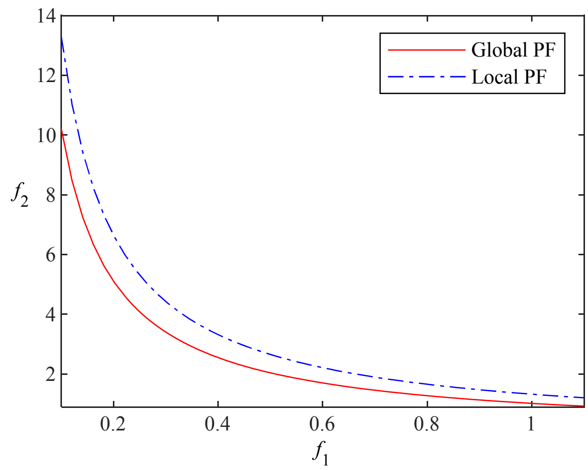
Fig. 1 shows the Local PS, Global PS, Local PF and Global PF of MMF11. As shown in Fig. 1 (a), x_1 and x_2 denote two dimensions of decision space and z-axis represents nondomination rank values. The smaller the rank value is, the better the solution is. There are two valleys in Fig. 1 (a). The deeper one represents the Global PS and the shallower is the local PS. In Fig. 1 (b), f_1 and f_2 are two dimensions of objective space. The solid curve is Global PF and the dashed line means local PF.

Definition 4 Multimodal Multiobjective Optimization Problem (MMOP): A multiobjective optimization problem is called MMOP if it has at least one local PS, or it has at least two Global PSs mapping to the same PF.

MMF11 shown in Fig. 1 is an MMOP with one Global PS and one local PS.



(a) Local PS, Global PS



(b) Local PF, Global PF

Fig. 1. The Local PS, Global PS, Local PF and Global PF of MMF11.

III. THE DIFFICULTIES OF SOLVING MMOPS WITH LOCAL PSS

It is not easy to solve MMOPs with local PSs. On one hand, solutions in local PS may be deleted. The reason is that solutions in local PS are likely to be dominated by those in Global PS. In Fig. 1, Global PS (the deeper valley in Fig. 1 (a)) maps to the solid line in Fig. 1 (b), while local PS (the shallower valley in Fig. 1 (a)) maps to the dashed line in Fig. 1 (b). The points in the dashed line are very likely dominated by the ones in the solid line. If the population is sorted according to nondominated relationship, the local Pareto optimal solutions are very likely deleted in the environmental selection process.

On the other hand, some algorithms may fall into local PS, thus missing Global PSs. In problems with a large and wide local valley, local PSs are very deceptive. If the search ability of an algorithm is not strong enough, it will fall into the local valley and cannot jump out.

Since there is little research on MMOPs with local PSs, this paper proposed an MMOGA for solving this kind of problem. The details of the proposed algorithm are described in the following section.

IV. Multimodal multiobjective genetic algorithm

In this section, the details of MMOGA are introduced and the effectiveness of this mechanism is analyzed.

A. Algorithm framework

Algorithm 1: Framework of MMOGA

Input: *pop* (population size), *N*_ops (number of PSs to be obtained), *NS* (neighborhood size)

1 //Initialization
2 Initialize Population *P* with *pop* individuals
3 Evaluate(*P*)
4 **while** *Generation* < *MaxGen* and *FEs* < *MaxFEs* **do**
5 //Determine neighborhood
6 **for** *i* = 1: *pop*
7 *P_neighbor*(*i*) = the individuals whose distance to *P*(*i*) is smaller than niching radium;
8 *Rank*(*i*) = the nondominated rank of *P*(*i*) in *P_neighbor*(*i*)
9 *CD*(*i*) = the mean distance from *P*(*i*) to individuals in *P_neighbor*(*i*)
10 //Select mate for *P*(*i*)
11 *Mate*(*i*) = the farthest nondominated individual in *P_neighbor*(*i*)
12 *Offspring*(*i*) = genetic_operator(*P*(*i*), *Mate*(*i*))
13 Get *rank_off*(*i*) and *CD_off*(*i*) in the same way as line 8 -9
14 // Update *P*(*i*)
15 Replace *P*(*i*) with *Offspring*(*i*) if *rank_off*(*i*) < *rank*(*i*) or [*rank_off*(*i*) = *rank*(*i*) & *CD_off*(*i*) > *CD*(*i*)]
16 **end for**
17 **end while**
18 //Output *N_ops* fronts
19 Final_P = select *N_ops* ranks in *P* (Delete the individuals dominated by its neighbors and select the first *N* fronts in the left individuals)
20 **Output**: Final_P

The main framework of MMOGA is illustrated in **Algorithm 1**. In MMOGA, the population evolves in local region and the nondominated solutions in each local region are selected. Then all the nondominated solutions are combined and ranked into different fronts. Finally, the first *N*_ops fronts are output.

In line 7 of **Algorithm 1**, niching radium is determined by *NS* (neighborhood size). The Euclidean distances between *P*(*i*) and its nearest *NS**pop* individuals are calculated in decision space. The niching radium is equal to the mean value of all the above distances. The individuals whose distance to *P*(*i*) are all its neighbors. The setting of *NS* is discussed in Section V.

Two indicators are used to judge the performance in each neighborhood. The first one is *Rank*(*i*) and the other is *CD*(*i*). *Rank*(*i*) is the nondominated rank value and *CD*(*i*) is the mean distance from *P*(*i*) to all its neighbors. Obviously, *Rank*(*i*) reflects the convergence of *P*(*i*). Individuals with low *Rank* have good convergence. *CD*(*i*) reflects the crowding degree in the neighborhood of *P*(*i*). Individuals with large *CD* have good diversity. These two indicators are used to determine whether *Offspring* can survive or not. *Offspring* can survive on the following two conditions (line 15): (1) the *Rank* of *Offspring* is lower than that of *P*; (2) the rank of *Offspring* is equal to that of *P* and its *CD* is larger than that of *P*.

In order to generate offspring with good convergence and diversity, individual with the lowest *Rank* and largest distance to *P*(*i*) is selected as its mate (line 11). *P*(*i*) and its mate generate offspring through genetic operator which is the same as the genetic algorithm.

After evolution, the population needs further selection because some surviving individuals are neither local Pareto optima nor global optima. In the further selection process, individuals are deleted once they are dominated by any of their neighbors. The left individuals are local or global Pareto solutions since they are not dominated by any of their neighbors. They are sorted into different ranks and the first *N*_ops ranks are output.

B. Algorithm analysis

This subsection explains why MMOGA can solve MMOPs with local PSs. In addition, the significances of keeping both global and local PSs are analyzed.

The decision niching scheme and environmental selection method enable MMOGA to find both local and global PSs. In decision niching scheme only individuals in the same neighborhood can mate and compete with each other. This scheme will prevent local Pareto optimal solutions from being dominated by global optimal ones. For example, in Fig. 2 the circles represent different neighborhoods in decision space. The neighborhoods with global and local solutions are denoted by solid circles. There is no overlap between them, so the solutions in these two circles can neither mate nor compete. They evolve generation by generation to find global and local PSs respectively. The disadvantage of this scheme is that the nondominated solutions in the circles without global or local PS will still be kept. These solutions are not desired. To deal with this problem, a further selection method is used to delete these undesired solutions. In the further selection process, the dominated solutions in every neighborhood are marked. All the marked solutions will be removed from the final population. Since there are overlaps among adjacent circles in Fig. 2, the dominated relationship can spread from one to the other. For example, the top part of circle A is dominated by the part at its bottom, so the top part is marked. A similar situation happens in circle B. Therefore, all the solutions in the circle without local and global PS are all marked. Only local or global solutions will be kept.

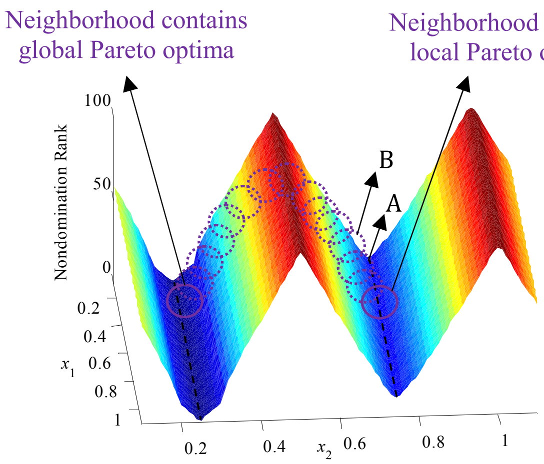


Fig. 2. The illustration of effective schemes in MMOGA.

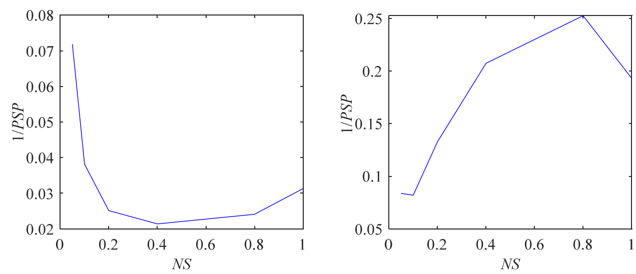
It is of great significance to keep both global and local PSs. First, it provides multiple choices for decision-makers. In some real-world applications, global Pareto optimal solutions are too expensive to obtained or they are not feasible anymore. Then, local Pareto optimal solutions are the best choices. Second, it helps reveal the potential characteristics of MMOPs. Many local Pareto optimal solutions are unknown until they are found. They can reflect the special characteristics.

The effectiveness of MMO algorithms is verified through experimental results in the next section.

V. Simulation Results

The MMO test suite in CEC2020 is used to test the effectiveness of the proposed algorithm. The test problems 1-15 have only global PSs while 16-24 have both global and local PSs. In the experiments, population size is set to 200**N*_ops and the maximal number of fitness evaluation is set to 10000**N*_ops according to the MMO technical report in CEC2020 [13] where *N*_ops represent the number of PSs to be obtained. The GA related parameters are the same as NSGAII [14]. All the experiments in the following texts are carried out 21 times. This section is organized as follows. First the setting of neighborhood size *NS* is discussed. Then the effectiveness of the niching method in decision space and the final selection method is verified. Third, the proposed algorithm is compared with state-of-the-art MMO algorithms. Finally, the computational complexity of the proposed algorithm is analyzed.

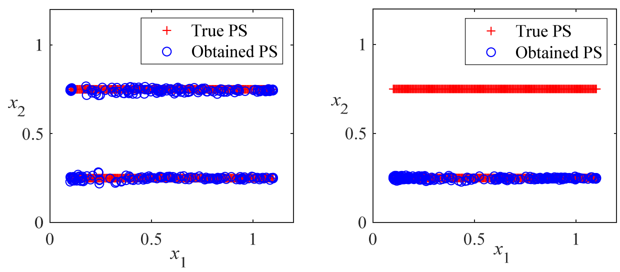
The neighborhood size *NS* is a key parameter in MMOGA. To set it properly, several experiments are carried out. The proposed algorithm using different *NS* values is tested on problems with 2-dimension and 3-dimension decision space separately. The indicator curves are shown in Fig. 3. MMF2 and MMF15_a_*l* are chosen as typical 2-dimension and 3-dimension problems. Fig. 3 (a) shows the indicators on MMF2. For 1/*PSP*, the smaller value means the better performance. As shown in Fig. 3 (a), the performance of MMOGA is improved as *NS* increasing from 0~0.4. However, when *NS* is larger than 0.4, the performance deteriorates. Its performance is relatively better when *NS* in [0.2, 0.6]. Therefore, the suggested setting of *NS* on 2-dimension problems is 0.2~0.6. In this paper, *NS* is set to 0.4 for 2-dimension problems. Fig. 3 (b) shows the indicators on MMF15_a_*l*. MMOGA's performance changes more than that on MMF2. It performs best when *NS* is equal to 0.1. Therefore, *NS* is set to 0.1 for 3-dimension problems.



(a) 1/*PSP* on MMF2 (b) 1/*PSP* on MMF15_a_*l*

Fig. 3. The influence of *NS* on algorithm performance.

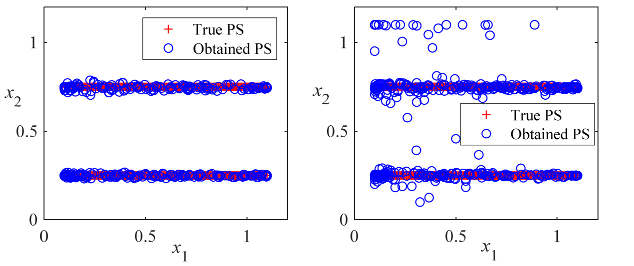
To verify the effectiveness of the niching method in decision space, a multiobjective genetic algorithm with and without niching method are compared on test function MMF11_*l*. The results are shown in Fig. 4, where MMOGA is the proposed algorithm and MOGA is a multiobjective genetic algorithm without the niching method. MMF11_*l* has one local PS (x_2 = 0.75, $x_1 \in$ [0.1,1.1]) and one global PS (x_2 = 0.25, $x_1 \in$ [0.1,1.1]). As shown in Fig. 4, MMOGA can find both global and local PSs while MOGA can find only the global one. It is because the local Pareto optimal solutions are compared with global Pareto optimal solutions in MOGA. Then the local Pareto optimal solutions are likely deleted since they are inferior to global Pareto optimal solutions. However, in MMOGA, the local Pareto optimal solutions are not compared with global Pareto optimal solutions directly. Instead, they are only compared with their neighbors. Therefore, MMOGA can find both local and global PSs.



(a) MMOGA (b) MOGA

Fig. 4. PSs obtained by MMOGA and MOGA

To verify the effectiveness of the final selection method, MMOGA with and without the final selection method are compared. The PSs obtained by MMOGA with and without the final selection method are shown in Fig. 5. In MMOGA, since individuals evolve in the local area, some of them may be misidentified as local optima. As shown in Fig. 5 (b), some circles (obtained Pareto optimal solutions) don't lie near crosses (true Pareto optimal solutions), which means they are not true local or global PS but be misidentified. Therefore, the survived individuals in each niche need combined and selected again. The final selection method marked all the individuals dominated by their neighbors and deleted all the marked ones. Then the individuals in the first *N*_ops rank are selected out. In Fig. 5 (a), the misidentified individuals are deleted, which can verify the effectiveness of the final selection method.

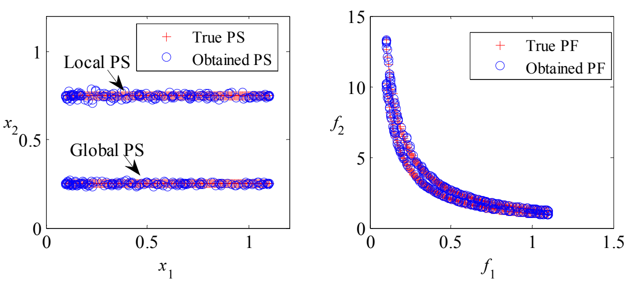


(a) MMOGA with final selection (b) MMOGA without final selection

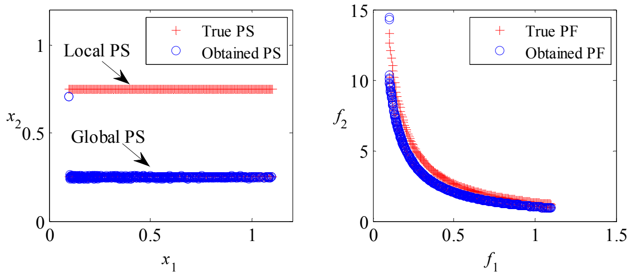
Fig. 5. PSs obtained by MMOGA with and without final selection method

Four algorithms, MMO-Clustering-PSO, DN-NSGAII, NSGAII and MO_Ring_PSO_SCD, are compared with MMOGA in this paper. Among them, MMO-Clustering-PSO is the champion in the multimodal multiobjective competition of CEC 2019. NSGAII is the most popular multiobjective genetic algorithm. DN-NSGAII is a modification of NSGAII for solving MMOPs. MO_Ring_PSO_SCD is a representative MMO algorithm.

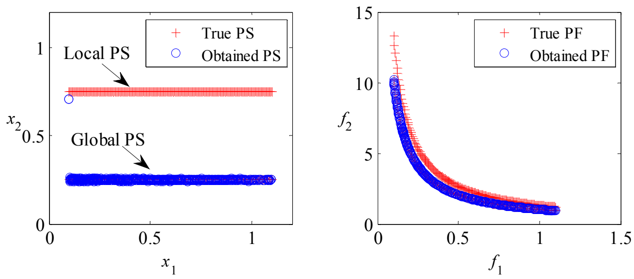
The obtained PSs and PFs with medium 1/*PSP* in 21 times are shown in Fig. 6 and Fig. 7. Fig. 6 shows the PSs and PFs of MMF11_*l*, which has one global and one local PS. The true PS is represented with the cross while the obtained PS is denoted with the circle. Among the two true PSs, the top one is local PS and the below one is global PS. As shown, DN-NSGAII and NSGAII can only find global Pareto optimal solutions because all the circles lie on the below true PS. MMO-Clustering-PSO and MO_Ring_PSO_SCD can find several local Pareto optimal solutions besides the global ones. MMOGA can find both global and local Pareto optimal solutions of MMF11_*l*. Fig. 7 shows the PSs and PFs of MMF13_*l*. MMF13_*l* has one global and one local PS, but they consist of several segments. The results are similar to those of MMF11_*l*. MMOGA can find most of the local and global Pareto optimal solutions. However, the distribution of the third segment is not very good, which needs to be further improved.



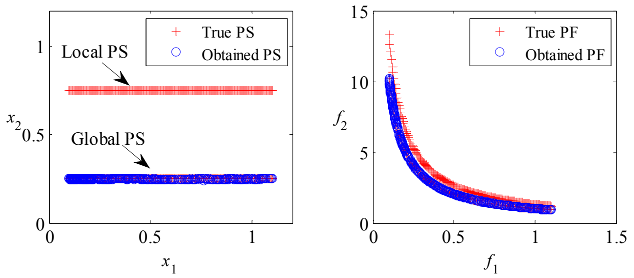
(a) PSs and PFs obtained by MMOGA



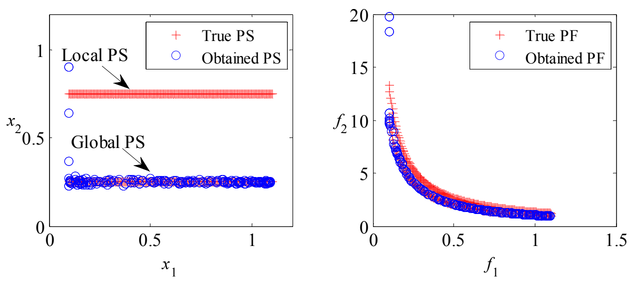
(b) PSs and PFs obtained by MMO-Clustering-PSO



(c) PSs and PFs obtained by DN-NSGAII

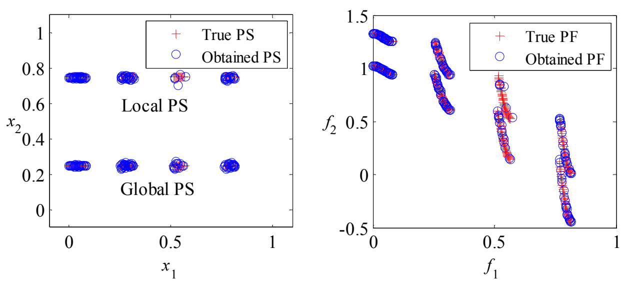


(d) PSs and PFs obtained by NSGAII

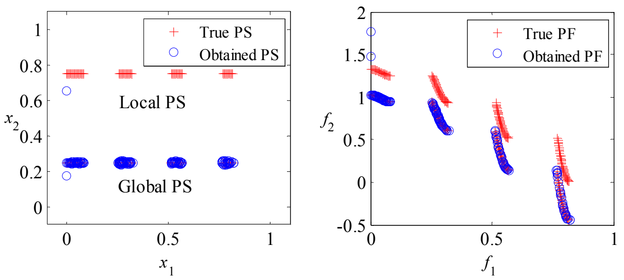


(e) PSs and PFs obtained by MO_Ring_PSO_SCD

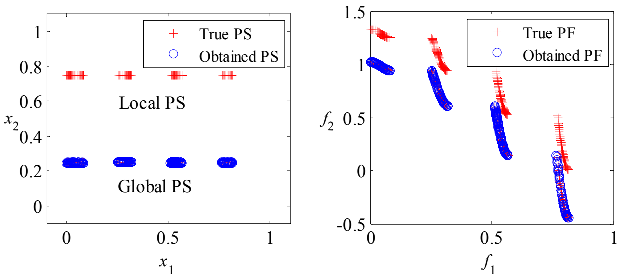
Fig. 6. The PSs and PFs of MMF11_*l*



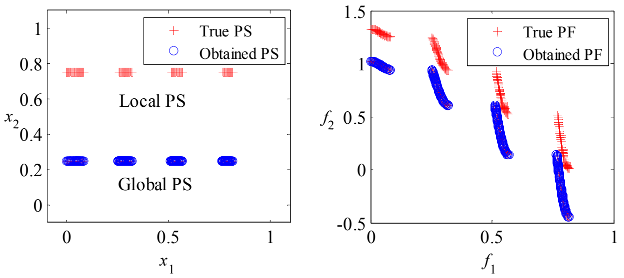
(a) PSs and PFs obtained by MMOGA



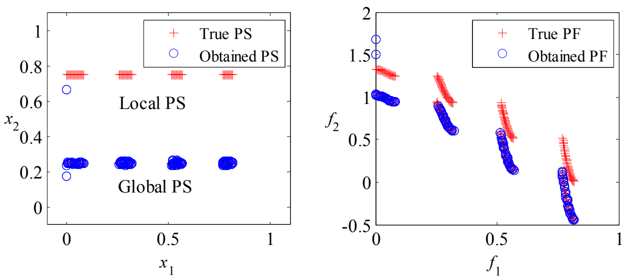
(b) PSs and PFs obtained by MMO-Clustering-PSO



(c) PSs and PFs obtained by DN-NSGAII



(d) PSs and PFs obtained by NSGAII



(e) PSs and PFs obtained by MO_Ring_PSO_SCD

Fig. 7. The PSs of MMF13_*l*

To compare these algorithms comprehensively, two performance indicators are used, 1/*PSP* and 1/*HV*. For 1/*PSP*, the smaller values mean the obtained PSs are closer to the true ones. For 1/*HV*, the smaller values mean the better performance in objective space. The indicators of MMOGA and comparison algorithms are shown in Table 1 and Table 2. The shadow rows represent test problems with both local and global PSs. The best results are in bold. As shown in Table 1, MMOGA performs best in all shadow problems, which demonstrates that MMOGA outperforms others on problems with both local and global PSs. Though it doesn't perform best in MMOPs with only global PSs (the non-shaded problems) according to 1/*PSP*, it performs best in these problems according to 1/*HV* (as shown in Table 2). According to No Free Lunch theorems, it cannot always perform best in every aspect. In conclution, the proposed algorihtm can solve MMOPs with local PSs and performs better in objective in problems with only global PSs.

The computational complexity of MMOGA mainly depends on the main loop of GA, nondominated rank and crowding distance calculation of parents and offspring. The computational complexity of the main loop is *O*(*pop*), where *pop* is the size of the population. Since there are *pop***NS* (*NS* is neighborhood size ranging from 0 to 1)individuals in each neighborhood, the rank and crowding distance calculation needs *pop* * *NS* * (*pop* * *NS* ± 1)/2 iterations. The nested computational complexity is *pop* * [*pop* * *NS* * (*pop* * *NS* ± 1)/2].

VI. Conclusion

This paper proposes a multimodal multiobjective genetic algorithm (MMOGA) to finding local and global PSs of MMOPs. In MMOGA, each individual is assigned a neighborhood according to the Euclidean distance in decision space. They can only mate and complete with individuals in their own neighborhoods. In this way, the population can evolve in different niches. However, some individuals may be misidentified as local Pareto optimal solutions. To eliminate these misidentified individuals, the nondominated solutions in each niche are combined. Any individual which is dominated by its neighbors is deleted. Experimental results verify that the proposed algorithm is effective in solving MMOPs with local PSs.

This work a preliminary study on finding both local and global PSs of MMOPs. Though the proposed algorithm performs better on MMOPs with local PSs, it is not as good as some MMO algorithms on MMOPs with only global PSs. In addition, the computational complexity is a little high. These shortcomings will be overcome in our future work. In addition, the performance of MMOGA on complex MMOPs, such as problems with narrow global PS but wide local PS, will be improved in the future.

References

[1] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," *Soft Computing,* vol. 17, no. 7, pp. 1283-1299, 2013.

[2] G. Zhang, X. Shao, P. Li, and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," *Computers & Industrial Engineering,* vol. 56, no. 4, pp. 1309-1318, 2009.

[3] C. T. Yue, J. J. Liang, B. Y. Qu, K. J. Yu, and H. Song, "Multimodal multiobjective optimization in feature selection," in *IEEE Congress on Evolutionary Computation*, 2019, pp. 302-309.

[4] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems,"

Evolutionary Computation, vol. 7, no. 3, pp. 205-230, 1999.

[5] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, Guanajuato, 2005, pp. 47-61: Springer.

[6] J. J. Liang, C. T. Yue, and B. Y. Qu, "Multimodal multi-objective optimization: A preliminary study," in *IEEE Congress on Evolutionary Computation*, Vancouver, 2016, pp. 2454-2461: IEEE, 2016.

[7] C. T. Yue, B. Y. Qu, and J. J. Liang, "A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems," *IEEE Transactions on Evolutionary Computation,* vol. 22, no. 5, pp. 805-817, 2018.

[8] Y. Liu, G. G. Yen, and D. Gong, "A Multi-Modal Multi-Objective Evolutionary Algorithm Using Two-Archive and Recombination Strategies," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 4, pp. 660-674, 2018.

[9] Y. Liu, H. Ishibuchi, and Y. Nojima, "A Double-Niched Evolutionary Algorithm and its Behavior on Polygon-Based Problems," in *International Conference on Parallel Problem Solving from Nature*, 2018, pp. 262-273.

[10] R. Tanabe and H. Ishibuchi, "A Decomposition-Based Evolutionary Algorithm for Multi-modal Multi-objective Optimization," in *International Conference on Parallel Problem Solving from Nature*, Coimbra, 2018, pp. 249-261: Springer.

[11] J. J. Liang *et al.*, "Multimodal multiobjective optimization with diffierential evolution," *Swarm and Evolutionary Computation,* vol. 44, no. 2019, pp. 1028-1059, 2018.

[12] C. T. Yue, B. Y. Qu, K. J. Yu, J. J. Liang, and X. Li, "A novel scalable test problem suite for multimodal multiobjective optimization," *Swarm and Evolutionary Computation,* vol. 48, no. 2019, pp. 62-71, 2019.

[13] J. J. Liang, P. N. Suganthan, B. Y. Qu, D. W. Gong, and C. T. Yue, "Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session on Multimodal Multiobjective Optimization," Zhengzhou University, 2020.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

Table 1 The 1/*PSP* of different algorithms (mean ± std)

Test function	MMOGA	MMO_Clustering_PSO	DN-NSGAII	NSGAII	MO_Ring_PSO_SCD
MMF1	0.0484 ± 0.0136	**0.0218 ± 0.0018**	0.0588 ± 0.0126	0.0640 ± 0.0047	0.0366 ± 0.0022
MMF2	0.0305 ± 0.0126	**0.0218 ± 0.0062**	0.0491 ± 0.0284	0.0573 ± 0.0067	0.0198 ± 0.0037
MMF4	0.0343 ± 0.0097	**0.0088 ± 0.0004**	0.0475 ± 0.0125	0.0609 ± 0.0125	0.0218 ± 0.0016
MMF5	0.0997 ± 0.0280	**0.0391 ± 0.0020**	0.1153 ± 0.0112	0.1291 ± 0.0106	0.0643 ± 0.0032
MMF7	0.0338 ± 0.0096	**0.0088 ± 0.0005**	0.0274 ± 0.0068	0.0460 ± 0.0065	0.0209 ± 0.0012
MMF8	0.2686 ± 0.1180	**0.0321 ± 0.0021**	0.1606 ± 0.0688	1.0021 ± 0.2526	0.0543 ± 0.0037
MMF10	0.1002 ± 0.0688	0.0320 ± 0.0186	0.1523 ± 0.1216	0.1533 ± 0.1131	**0.0254 ± 0.0081**
MMF11	0.0151 ± 0.0044	**0.0033 ± 0.0001**	0.0046 ± 0.0003	0.0040 ± 0.0006	0.0082 ± 0.0009
MMF12	0.0120 ± 0.0116	0.0032 ± 0.0004	**0.0027 ± 0.0009**	0.0030 ± 0.0003	0.0045 ± 0.0005
MMF13	0.0730 ± 0.0207	**0.0316 ± 0.0041**	0.0772 ± 0.0173	0.1293 ± 0.0500	0.0478 ± 0.0036
MMF14	0.0569 ± 0.0162	**0.0240 ± 0.0004**	0.0860 ± 0.0083	0.1121 ± 0.0164	0.0511 ± 0.0014
MMF15	0.0749 ± 0.0211	**0.0273 ± 0.0007**	0.0781 ± 0.0053	0.0838 ± 0.0075	0.0555 ± 0.0020
MMF1_e	0.5883 ± 0.2555	0.4872 ± 0.1847	0.9457 ± 0.4894	2.3378 ± 0.7170	**0.4178 ± 0.1541**
MMF14_a	0.0617 ± 0.0175	**0.0281 ± 0.0005**	0.1033 ± 0.0077	0.1175 ± 0.0060	0.0583 ± 0.0017
MMF15_a	0.0745 ± 0.0209	**0.0313 ± 0.0011**	0.1061 ± 0.0130	0.0985 ± 0.0069	0.0617 ± 0.0026
MMF10_*l*	**0.0300 ± 0.0139**	0.1743 ± 0.0099	4.0043 ± 3.4689	5.8494 ± 4.3221	0.1646 ± 0.0080
MMF11_*l*	**0.0130 ± 0.0082**	0.6452 ± 0.5943	2.0136 ± 0.1698	3.3077 ± 0.5348	0.3320 ± 0.2947
MMF12_*l*	**0.0049 ± 0.0016**	0.6194 ± 0.5762	2.6715 ± 0.1746	4.5955 ± 0.7606	0.4439 ± 0.4451
MMF13_*l*	**0.2566 ± 0.0982**	0.3782 ± 0.1036	0.5963 ± 0.0208	0.6827 ± 0.0363	0.3814 ± 0.0988
MMF15_*l*	**0.0579 ± 0.0027**	0.1542 ± 0.0157	0.2785 ± 0.1100	0.4023 ± 0.1842	0.1567 ± 0.0256
MMF15_a_*l*	**0.0750 ± 0.0042**	0.1607 ± 0.0139	0.2279 ± 0.0335	0.2860 ± 0.0063	0.1744 ± 0.0255
MMF16_*l*1	**0.0623 ± 0.0047**	0.0979 ± 0.0083	0.2074 ± 0.0300	0.2082 ± 0.0300	0.1110 ± 0.0073
MMF16_*l*2	**0.0980 ± 0.0076**	0.2078 ± 0.0230	0.5141 ± 0.1959	0.6006 ± 0.1861	0.2035 ± 0.0206
MMF16_*l*3	**0.1189 ± 0.0088**	0.1429 ± 0.0087	0.2838 ± 0.0343	0.3148 ± 0.0043	0.1458 ± 0.0106

Table 2 The 1/*HV* of different algorithms (mean ± std)

Test function	MMOGA	MMO_Clustering_PSO	DN-NSGAII	NSGAII	MO_Ring_PSO_SCD
MMF1	**1.0607 ± 0.31147**	1.1435 ± 0.0003	1.1447 ± 0.0006	1.1435 ± 0.0002	1.1466 ± 0.0004
MMF2	**1.0786 ± 0.31483**	1.1635 ± 0.0036	1.1577 ± 0.0105	1.1570 ± 0.0053	1.1648 ± 0.0037
MMF4	**1.7235 ± 0.50579**	1.8462 ± 0.0007	1.8489 ± 0.0004	1.8466 ± 7.3034e-05	1.8550 ± 0.0011
MMF5	**1.0619 ± 0.31172**	1.1435 ± 0.0002	1.1443 ± 0.0007	1.1437 ± 0.0006	1.1461 ± 0.0003
MMF7	**1.0633 ± 0.31198**	1.1424 ± 8.9383e-05	1.1450 ± 0.0004	1.1432 ± 0.0002	1.1458 ± 0.0003
MMF8	**2.2636 ± 0.65931**	2.3754 ± 0.009	2.3690 ± 0.0016	2.3632 ± 0.0002	2.3927 ± 0.0106
MMF10	**0.0770 ± 0.0222**	0.0799 ± 0.0007	0.0821 ± 0.0030	0.0815 ± 0.0029	0.0803 ± 0.0006
MMF11	**0.0645 ± 0.0189**	0.0689 ± 1.9573e-05	0.0689 ± 1.4135e-05	0.0689 ± 6.2838e-06	0.0692 ± 6.0029e-05

MMF12	**0.6012 ± 0.1736**	0.6389 ± 0.0026	0.6362 ± 0.0011	0.6356 ± 0.0001	0.6398 ± 0.0018
MMF13	**0.0512 ± 0.0149**	0.0544 ± 3.3044e-05	0.0543 ± 8.6362e-05	0.0542 ± 3.2812e-06	0.0547 ± 7.8610e-05
MMF14	**0.2925 ± 0.0836**	0.3167 ± 0.0261	0.3274 ± 0.0129	0.3518 ± 0.0021	0.3444 ± 0.0172
MMF15	**0.2080 ± 0.0588**	0.2322 ± 0.0116	0.2318 ± 0.0109	0.2343 ± 0.0046	0.2443 ± 0.0130
MMF1_e	**1.1945 ± 0.3487**	1.1582 ± 0.0061	1.1664 ± 0.0203	1.1567 ± 0.0031	1.1703 ± 0.0141
MMF14_a	**0.2947 ± 0.0837**	0.3334 ± 0.0295	0.3142 ± 0.0087	0.3526 ± 0.0074	0.3314 ± 0.0214
MMF15_a	**0.2099 ± 0.0594**	0.2262 ± 0.0085	0.2391 ± 0.0138	0.2435 ± 0.0049	0.2424 ± 0.0123
MMF10_*l*	**0.0781 ± 0.0226**	0.0789 ± 0.0003	0.0798 ± 0.0025	0.0803 ± 0.0030	0.0796 ± 0.0005
MMF11_*l*	**0.0654 ± 0.0191**	0.0688 ± 9.5625e-06	0.0688 ± 4.9943e-06	0.0688 ± 2.1817e-06	0.0690 ± 2.4963e-05
MMF12_*l*	0.7407 ± 0.2123	0.6373 ± 0.0011	0.6355 ± 0.0004	**0.6352 ± 2.2507e-05**	0.6373 ± 0.0004
MMF13_*l*	**0.0507 ± 0.0148**	0.0543 ± 1.4776e-05	0.0542 ± 7.6699e-06	0.0542 ± 1.0033e-06	0.0545 ± 4.7165e-05
MMF15_*l*	0.2387 ± 0.0094	**0.2243 ± 0.0081**	0.2303 ± 0.0094	0.2391 ± 0.0038	0.2385 ± 0.0167
MMF15_a_*l*	0.2472 ± 0.0134	**0.2243 ± 0.0064**	0.2259 ± 0.0106	0.2332 ± 0.0037	0.2386 ± 0.0103
MMF16_*l*1	0.2342 ± 0.0109	0.2262 ± 0.0080	**0.2250 ± 0.0061**	0.2337 ± 0.0019	0.2288 ± 0.0117
MMF16_*l*2	0.2346 ± 0.0106	**0.2212 ± 0.0053**	0.2281 ± 0.0059	0.2343 ± 0.0021	0.2325 ± 0.0115
MMF16_*l*3	0.2257 ± 0.0122	0.2246 ± 0.0078	**0.2236 ± 0.0059**	0.2353 ± 0.0013	0.2340 ± 0.0118