Information Sciences 581 (2021) 304–326

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A cluster-based immune-inspired algorithm using manifold learning for multimodal multi-objective optimization

Weiwei Zhang [a], Ningjun Zhang [a], Weizheng Zhang [a], Gary G. Yen [b,*], Guoqing Li [c]

[a] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China
[b] School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA
[c] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

ARTICLE INFO

Article history:
Received 30 January 2021
Received in revised form 11 September 2021
Accepted 15 September 2021
Available online 17 September 2021

Keywords:
Multimodal multi-objective optimization
Immune-inspired algorithm
Manifold learning
Principal component analysis

ABSTRACT

Both problem characteristics in multimodality and multi-objective are involved in multimodal multi-objective optimization problems (MMOPs). How to locate diverse Pareto sets and approximate Pareto front simultaneously is a challenging research topic. To address this issue, a cluster-based immune-inspired algorithm using manifold learning is proposed in this paper for solving MMOPs. First of all, the population is partitioned into multiple subpopulations, and each of them is expected to find equivalent Pareto solutions in different regions. Subsequently, the immune-inspired algorithm with proportional cloning and hypermutation is developed for improving the diversity of the population and obtaining high-quality Pareto solutions in the decision space. Additionally, principal component analysis is adopted to learn the manifold of the Pareto set, further improve the convergence, and enhance interaction among subpopulations. The proposed algorithm is compared with six state-of-the-art algorithms. Experimental results demonstrate that the proposed algorithm is capable of locating equivalent Pareto optimal solutions in the decision space and maintaining the diversity and convergence of solutions in both decision space and objective space, simultaneously.

1. Introduction

Multi-objective optimization problems (MOPs) are commonly seen in real-world applications, such as industrial scheduling[1], environmental/economic dispatch[2], fighter vehicle control[3], and signal reconstruction[4], to name a few. Multiple conflicting objectives are involved in MOPs to be optimized simultaneously. For instance, in the manufacturing industry, spare parts purchasing is a crucial task to manufacturing firms. Price and quality are commonly considered when purchasing the spare parts. Finding adequate spare parts with high quality and low price is a vital issue for decision makers. However, better quality usually implies a higher price, i.e., price and quality are two conflicting objectives. Improving upon one objective always leads to degradation on another. Over the past two decades, a large number of multi-objective evolutionary algorithms (MOEAs) have been proposed to solve MOPs[5,6,7,8]. Generally, MOEAs are intended to find diverse Pareto optimal solutions with excellent distribution and converge to the Pareto front (*PF*). In real-world applications, many MOPs exhibit multimodality characteristics, i.e., there are multiple Pareto optimal solutions (PSs) corresponding to the same objective values in MOPs, such as space mission design problems [9] and functional brain imaging problems[10]. This kind of MOPs is

[*] Corresponding author.
E-mail address: gyen@okstate.edu (G.G. Yen).

https://doi.org/10.1016/j.ins.2021.09.043

termed multimodal multi-objective optimization problems (MMOPs). Continuing on the spare parts example[11], different suppliers provide spare parts with varying qualities and prices based on the adopted materials and manufacturing processes. There obviously exists a PF when considering the spare parts purchasing problem as a multi-objective optimization problem in which both price and quality are the considered objectives. Due to market operation or other reasons, different spare parts come from different suppliers with different materials and techniques may have the same quality and price. That is to say, to an optimal solution in the PF, there are more than one solution in the PS (multiple suppliers could produce the spare parts with the same price and quality by using different materials or techniques). Assuming that a Pareto optimal solution becomes infeasible due to environmental or practical constraints, finding multiple *PSs* in this scenario is necessary for offering decision-makers more feasible alternatives. As an illustrating example shown in Fig. 1, there are nine distinct Pareto subsets (i.e., PS_1 to PS_9) mapping into the same Pareto front (i.e., PF). In this example, the Pareto Set PSis the union of all the different subsets PS = $\bigcup_k \mathrm{PS}_k$, $k = 9$, where PS_kbe the k*th* group of Pareto optimal solutions, and the Pareto set PS is the union of all the PS_k[12].

MOPs concentrate on finding high-quality Pareto optimal solutions in the objective space, while MMOPs pay attention to both the multimodality in the decision space and multiple objectives in the objective space. Given the effectiveness within the framework of single-objective optimization, niching technique[13], diversity maintenance[14], and multiple population strategy[15] were naturally adopted to handle multimodality when coping with MMOPs. Therefore, integrating multimodal techniques with multi-objective evolutionary algorithms is an intuitive methodology for solving MMOPs [16]. Additionlly, several nature-inspired algorithms, i.e., immune-inspired algorithms, with the inherent ability of diversity maintenance and locating multiple local optima bear the appreciable advantages of handling the multimodality as well. Therefore, integration of both would have great potential to handle the multimodality in MMOPs. In this paper, the multiple population strategy and immune-inspired algorithm are seamlessly integrated to find multiple equivalent Pareto optimal solutions for solving MMOPs.

On the other hand, many algorithms promote the diversity of PSs in the decision space[17], but they have traded the performance of Pareto optimal solutions in the objective space. Therefore, it is a crucial issue to locate Pareto optimal solutions set as diverse as possible in the decision space and improve the convergence of solutions in the objective space, simultaneously. Under a certain condition, it is proven that the PS of a continuous MOP with m objectives is a (m-1)-dimensional manifold in the decision space[18]. Manifold learning techniques[19] are capable of finding several useful low-dimensional manifold structures hidden in high-dimensional space. On the basis of this fact, building a model from the structures and sampling technique, the manifold learning strategy can produce high-quality solutions to approach the true PF [19]. In this paper, principal component analysis (PCA) is adopted to explore the manifold structures from the neighborhood of the subpopulations to generate high-quality solutions. Moreover, it also enhances the interaction among subpopulations and accelerate population convergence.

The main contributions of this paper are summarized as follows:

1. K-means clustering technique is introduced to partition the population into multiple subpopulations. Multiple subpopulations are responsible for finding equivalent Pareto optimal solutions in different regions in the decision space.
2. Immune-inspired algorithm with proportional cloning and hypermutation is devoted to each subpopulation. Particularly, the proportional cloning operator modulates the distribution of population while the hypermutation is favorable to improve the quality of solutions.
3. To improve the convergence performance of solutions, principal component analysis is called upon in the proposed algorithm to learn the manifold structures from the neighborhood of the subpopulations.

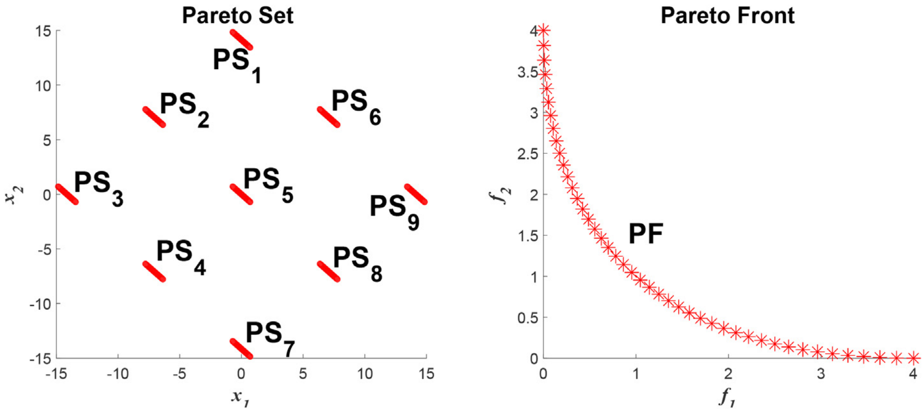


Fig. 1. The red circles represent nine distinct Pareto subsets (i.e., PS_1 to PS_9) corresponding to the same Pareto front (i.e., PF).

The rest of the paper is organized as follows. Section 2 describes the framework of the immune-inspired algorithm, estimation of distribution algorithm, and the related works on MMOPs. The proposed cluster-based immune-inspired algorithm using manifold learning for multimodal multi-objective optimization (in short MMOIM) is presented in Section 3. In Section 4, experimental analysis is performed. The conclusions and future work are presented in Section 5.

2. Related works

In this section, we introduce the relevant preliminary knowledge, including immune-inspired algorithm, estimation distribution algorithm, and manifold learning. The classic theory, operators, and algorithms involved in this paper are presented, and followed by briefly reviewing the previous works on multimodal multi-objective optimization.

2.1. Immune-inspired algorithm

Inspired by the clonal selection principle, the immune-inspired algorithm is proposed to tackle some complex optimization problems. **Algorithm 1** presents the main framework of immune-inspired algorithm.

```
Algorithm 1 Immune-inspired algorithm
1. Initialization: Randomly initialize antibody population.
2. Evaluation: Evaluate the objective values of the antibody population as their fitness.
3. Cloning: Generate copies of the antibodies.
4. Hypermutation: Mutate all the generated copies.
5. Selection: Select the one with highest fitness to survive.
6. Repeat steps 2–5 until a termination criterion is met.
```

To tackle multi-objective optimization problems, three main operators are involved in immune-inspired algorithms, including Cloning, Hypermutation, and Selection[20].

Cloning. According to the clonal selection principle, the antibodies with higher antibody-antigen affinity are capable of producing multiple clones. There is a variety of options to perform the cloning selection in real-world applications[21]. Among them, the proportional cloning operation[20] is one of the most popular methodologies. Assuming that $A_t = \{ab_1, ab_2, \cdots, ab_{Nab}\}$ is a population to conduct the cloning operator, where Nab denotes the antibodies number. The proportional cloning operation T^C is defined as follows:

$$\begin{aligned} T^C(ab_1, ab_2, \cdots, ab_{Nab}) &= \left\{T^C(ab_1), T^C(ab_2), \cdots, T^C(ab_{Nab})\right\} \\ &= \left\{ \underbrace{\{ab_1, ab_1, \cdots, ab_1\}}_{q_1}, \underbrace{\{ab_2, ab_2, \cdots, ab_2\}}_{q_2}, \cdots, \underbrace{\{ab_{Nab}, ab_{Nab}, \cdots, ab_{Nab}\}}_{q_{Nab}} \right\} \end{aligned} \tag{1}$$

where $T^C(ab_j) = q_j \times ab_j, j = 1, \cdots, Nab$, and q_j refers to number of each antibody $\{ab_1, ab_2, \cdots, ab_{Nab}\}$. Here, q_j is calculated as follows:

$$q_j = \left\lceil CS \times \frac{cd(ab_j)}{\sum_{k=1}^{Nab} cd(ab_k)} \right\rceil \tag{2}$$

where CS is the clonal scale, $cd(ab_j)$ indicates the crowding-distance value[20] of the jth antibody ab_j, and $\lceil\cdot\rceil$ is the ceiling operaor.

Hypermutation. Distinct from other evolutionary algorithms, there is no crossover operation in the clonal selection algorithm. Instead, each antibody performs the hypermutation after cloning. Hypermutation plays a critical role in maintaining the diversity of species. According to the clonal selection principle, the mutation rate is inversely proportional to antibody-antigen affinity. There are several ways to conduct the hypermutation operator, including static hypermutation, proportional hypermutation, inversely proportional hypermutation[22,23]. The hypermutation operator $T^H$is defined as

$$T^H(ab_1, ab_2, \cdots, ab_{Nab}) = \left\{T^H(ab_1), T^H(ab_2), \cdots, T^H(ab_{Nab})\right\} \tag{3}$$

Selection. Generally, the antibodies with higher affinity are chosen to survive. There are several methods to select some promising solutions for the next generation according to the requirement of the underlying optimization tasks.

2.2. Estimation of distribution algorithms and manifold learning

Estimation of distribution algorithm (EDA) is a special evolutionary paradigm. It generates new solutions through statistical learning, built a probabilitic modal, and sampling, instead of the traditional crossover or mutation used in memetic computing algorithms or hyper-heuristics[22]. Based on the Karush-Kuhn-Tucker condition, the Pareto set of multi-objective optimization problems is a piecewise continuous $(m-1)$-dimensional manifold in the decision space, where m is the number of objective functions[18]. It would be a clear benefit for the design of the algorithm if the regularity property is validated. Manifold learning is a typical technique to uncover regularity. Manifold learning algorithm is a kind of dimensionality reduction technique, and it has the capacity to transform high-dimensional data into a low-dimensional space. Numerous manifold learning algorithms have been introduced into evolutionary algorithms, including principal component analysis (PCA), locally linear embedding (LLE), multidimensional scaling (MDS), covariance pattern search [24,25,26] and so on. The most representative regularity model-based multi-objective EDA is RM-MEDA proposed by Zhang[18]. Since then, several hybrid EDAs were proposed with the regularity model to solve multi-objective optimization problems. RM-MEDA approximates the whole nonlinear PS through establishing several (m-1)-dimensional linear manifolds distribution models by PCA, and then sampling the new individuals from these models.

$$\xi = \psi_1 + \psi_2 + \cdots + \psi_K + \varepsilon \tag{4}$$

where $\psi_j, j = 1, 2, \cdots, K$, are the linear probabilistic models and ε is a noise vector. The union of linear probabilistic models $\psi_j, j = 1, 2, \cdots, K$ may provide a good approximation to the PS. **Algorithm 2** presents the main procedure of RM-MEDA.

The smallest hyper-rectangle containing the projections of all the points of S^i in the affine $(m-1)$-dimensional principal subspace of S^i is

$$\phi^i = \left\{ x \in R^D \middle| x = \bar{x}^i + \sum_{i=1}^{m-1} \alpha_k U_k^i, a_k^i \leqslant \alpha_k \leqslant b_k^i, k = 1, \cdots, m-1 \right\} \tag{5}$$

where $a_k^i = \min\limits_{x \in S^i} \left(x - \bar{x}^i\right)^T U_k^i$, $b_k^i = \max\limits_{x \in S^i} \left(x - \bar{x}^i\right)^T U_k^i$, α_k is a random number between a_k^i and b_k^i, and D is the dimension of the decision space.

And, set

$$\psi^i = \left\{ x \in R^D | x = \bar{x}^i + \sum_{k=1}^{m-1} \alpha_k U_k^i, a_k^i - 0.25\left(b_k^i - a_k^i\right) \leqslant \alpha_k^i \leqslant b_k^i + 0.25\left(b_k^i - a_k^i\right), i = 1, \cdots, m-1 \right\} \tag{6}$$

Each individual can be considered as an independent observation of the probabilistic model:

$$x = x' + \varepsilon' \tag{7}$$

where $\varepsilon' \in N(0, \sigma^i I)$ is a noise vector, and $\sigma^i = \frac{1}{D-m+1}\sum_{i=m}^{D} \lambda_j^i$, λ_j^i is the *kth* largest eigenvalue of the covariance matrix of the points in S^i, I is the $D \times D$ identity matrix.

Algorithm 2 Pseudo-code of RM-MEDA

```
Input: the number of the linear probabilistic models K
Output: P(t + 1)
Generate an initial population P(t) ;
Set t = 0 ;
while the stopping criterion is not met do
        //Modeling process
    Divide the population P(t) into K clusters, $S^1$, $S^2$, $\cdots S^K$, by local PCA;
    Let $\bar{x}^i$ be the mean of cluster $S^i$ and $U_k^i$ be the kth principal component, build probability model for each cluster $S^i$ by (5)
    Extend $\phi^i$ by 50% along each direction $U_k^i$ to get $\varphi^i$ by (6)
        //Reproduction
    Randomly sample a point $x'$ from $\varphi^i$ uniformly, $i = 1, 2, \cdots, K$ and generate new solutions as (7)
    Generate a new solution set Q by (7);
        //Selection
    Select |P(t)| individuals to form the new population |P(t + 1)| from P(t) ∪ Q, where |.| represent the cardinality of the set;
    t = t + 1 ;
end while
return the non-dominated individuals P(t + 1)
```

2.3. Previous works on multimodal multi-objective optimization

In recent years, researchers have paid attention on decision space for locating PSs and accelerated the development of researches on MMOPs. A review of evolutionary multi-modal multi-objective optimization was published[27]. Particularly, five distinct issues on multimodal multi-objective evolutionary algorithms (MMEAs) were surveyed: l) Pareto dominance-based MMEAs; 2) Decomposition-based MMEAs; 3) Set-based MMEAs; 4) A post-processing approach; 5) and Open issues.

The traditional non-dominated sorting faces a serious challenge in solving MMOPs since only the distribution of Pareto optimal solutions in the objective space was considered. Therefore, the non-dominated sorting methods were redefined in multi-objective evolutionary algorithms (MOEAs). Xia[28] proposed two unique crowding estimation methods to maintain diversity in both decision space and objective space. Omni-optimizer[29] was originated from NSGAII, and the concept of crowding distances for the decision space was first introduced in Omni-optimizer to solve various problems. To cope with MMOPs, Yue[30] proposed a special crowding distance based on the previous works, which reduced the negative effect from boundary solutions.

Since MMOPs were derived from multi-objective optimization involving multimodality problem characteristics, the combination of the multimodal strategy and multi-objective optimization method would present an intuitive solution to the underlying problems. As one of the representative methods, niching strategy was frequently used to combine with multi-objective evolutionary algorithms. In DN-NSGAII[31], a niching strategy was combined with a fast nondominated sorting method. Additionlly, a distance-based selection operator was applied to select Pareto optimal solutions. In MO_Ring_PSO_SCD[30], a niching strategy was combined with special crowding distance and non-dominant sorting. And, an index-based ring topology was used to induce stable niches. In NIMMO[8], the niching strategy was combined with an indicator-based approach. Here, the fitness calculation was performed among a child and its closest neighbor in solution space to maintain the diversity. In MMO-MOES [32], niching strategy is adopted to preserve uniform distances from each other and spread to the whole Pareto set automatically. Moreover, multi-population played an important role in coping with multimodality. Zhang[18] designed a new decision variable clustering method based on PSO with leader updating mechanism and ring-topology (MMO-CLRPSO) for MMOPs. In MMO-CLRPSO, the global-best PSO and local-best PSO with ring topology were ensembled to realize independent evolution of subpopulations and information interaction, respectively.

On the other hand, inspired by decomposition-based multi-objective evolutionary algorithms, several decomposition-based multimodal multi-objective evolutionary algorithms were developed to solve MMOPs. A MOEA/D variant[16] was used to solve MMOPs, in which several solutions were assigned to each weight vector to maintain diversity in the decision space. Tanabe[33] proposed a decomposition-based multimodal multi-objective evolutionary algorithm, called MOEA/D-AD. In MOEA/D-AD, multiple individuals which were far from each other in the solution space were assigned to the same decomposed single-objective subproblem, and only similar individuals in the solution space were compared based on their scalarizing function values. In addition, the decomposition-based evolutionary strategy was also applied in multimodal neighborhood-sensitive archived evolutionary many-objective optimization algorithm (MM-NAEMO)[34] and multimodal multi-objective optimization using a density-based one-by-one update strategy (MMOEA-GD) [35].

Some researchers employed machine learning to excavate the mapping relation of the decision space and the objective space to guide the evolutionary procedures. Liang et al. designed SMPSO-MM[36] where the self-organizing map was utilized to find the distribution structure of the population and to build the neighborhood in the decision space. Hu proposed MMOPIO[7] in which the self-organizing map (SOM) was combined with the improved pigeon-inspired optimization (PIO) for better control of the decision space, and thus, contributed to building a good neighborhood relation for the improved PIO. Qu proposed an algorithm named SS-MOPSO[37] to search for multiple Pareto optimal solutions.

In addition, the archive-based algorithm has proven to be effective as well. Liu presented a competitive algorithm termed as TriMOEA-TA&R[6] in which two-archive and recombination strategies were introduced. Maity et al. built upon the original framework of NAEMO[38] proposed multimodal NAEMO (MM-NAEMO) [34]. Although some appreciable achievements have been made, the existing algorithms always suffer from low accuracy of convergence and nonuniform distribution. That is to say, some of the obtained solutions tend to be near, but not completely converge to the true PSs and PF, and only a number of disperse parts of the PSs and PF are covered. In MOP, it has been proven the manifold learning can produce high-quality solutions to guide the population to approximate the real Pareto surface. Therefore, a manifold learning strategy is adopted here to accelerate the convergence in the paper. Moreover, the multi-population strategy and immune-inspired algorithm are employed for achieving the uniform distribution of the solutions and locating multiple local optima.

3. Proposed method

In this section, a cluster-based immune-inspired algorithm exploiting manifold learning, called MMOIM, is proposed for solving multimodal multi-objective optimization. MMOIM is composed of three essential components, including the multiple population strategy for multimodality, proportional cloning and hypermutation mechanism in the immune-inspired algorithm for subpopulation evolution, and the multi-population based manifold learning strategy for local optimization.

3.1. Multiple population strategy

A population *pop* with N antibodies is randomly initialized at first. Then, the antibody is separated into Ns subpopulations $\{subpop_i\}$, $i = 1, \cdots, Ns$ according to k-means cluster, and each subpopulation $subpop_i$ contains a number of antibodies $\{ab_j^i\}$, $j = 1, \cdots, Nab$, where Nab is the number of antibodies in the $subpop_i$.

3.2. Immune-inspired algorithm

The proposed proportional cloning and hypermutation based immune algorithm is shown in **Algorithm 3**. Each antibody ab_j^i, $j = 1, \cdots, Nab$, in the subpopulation $subpop_i$ produces q_j^i clones according to the proportional cloning operation [20] by (4), where the clone number q_j^i is defined as follows.

$$q_j^i = \left\lceil CS \times \frac{cd\left(ab_j^i, subpop_i\right)}{\sum_{h=1}^{Nab} cd\left(ab_h^i, subpop_i\right)} \right\rceil \tag{8}$$

where CS is a user-defined constant to determine the clonal scale, $cd\left(ab_j^i, subpop_i\right)$ represents the crowding-distance value of the jth antibody ab_j^i. Afterward, the static hypermutation [23] is applied on each of the clones as (3), in which each element of the variable vector is changed by a general mutation operator with probability p_m. To save the computational resource, most Max_{subpop} mutated antibodies survive in each subpopulation. There are several methods available to delete redundant antibodies. Considering the degree of the crowdedness, the harmonic average distance [35] is chosen. The antibodies with lower harmonic average distance are maintained, while the othes are deleted.

Algorithm 3 Pseudo-code of proportional cloning and hypermutation based immune algorithm

```
Input: subpopulation $subpop_i$ , clonal scale$CS$
Output: $subpop_i$
for each antibody $ab_j^i \in subpop_i$
    Generate $q_j^i$ clones according to the proportional cloning operation;
    Hypermutate each clone through static hypermutation;
end for
if the size of mutated clones is larger than $Max_{subpop}$
    Keep the $Max_{subpop}$ mutated clones with the largest harmonic average distance, and delete the others;
end if
Evaluate the fitness of the mutated clones;
Put the mutated clones into subpopulation $subpop_i$;
return the subpopulation $subpop_i$ ;
```

3.3. Manifold learning strategy

Generally, there are two stages in manifold learning, including modeling and sampling. In the modeling process, the probability model is built based on the distribution of the current solutions. Then in the sampling process, the new solutions are generated by adding the noisy disturbance to the sampling point from the built model. In our implementation, the manifold learning process[18] is applied to each subpopulation $subpop_i$, $i = 1, \cdots, Ns$, to approximate the distribution. The center, $Center_i$, of each subpopulation in the decision space is calculated at first. Then, the two nearest neighbors $\{subpop_{i-1}, subpop_{i+1}\}$ of subpopulation $subpop_i$ are determined according to the Euclidian distance among their centers. Hereafter, the antibodies in the union of the subpopulations $subPOP_i = \{subpop_{i-1}, subpop_i, subpop_{i+1}\}$ are used for building the probabilitic modal. There are several manifold learning methods available to build the modal. For simplicity, the PCA adopted in RM-MEDA[18] is applied here.

Algorithm 4 Pseudo-code of multi-population manifold learning strategy

```
Input: subpopulation $subpop_i$, $i = 1, \cdots, Ns$, Sampling scale $Ss$
Output:$subpop_{\mathrm{i}}$
Calculate the center $Center_i$ of each subpopulation $subpop_i$, $i = 1, \cdots, Ns$
for $i = 1, \cdots, Ns$
  Find two nearest neighbors $\{subpop_{i-1}, subpop_{i+1}\}$ of $subpop_i$
    $subPOP_i \leftarrow \{subpop_{i-1}, subpop_i, subpop_{i+1}\}$
  Compute the mean $\overset{-\ i}{ab}$ and the principle component $U_i^k$ of $subPOP_i$, and build the probability modal as Eq.(5) and get
  the final distribution $\psi_i$ by Eq.(9)
  for $j = 1, \cdots, Ss$
    //Sampling
    Uniformly randomly generate a point $ab'$ from $\psi_i$
    Generate a noise vector $\varepsilon'$ from$N(0, \sigma_i I)$
    Produce a new individual $ab_j$ by Eq. (9)
    $subpop_i \ = \{subpop_i, ab_j\}$
  end for
end for
```

Compute the mean $\bar{ab}^i$ and the principal component U_i^k of $subPOP_i$, and build the probability modal as (5) and obtain the final distribution ψ^i by (6). After that, Ss new solutions are generated by (9),

$$ab = ab' + \varepsilon' \tag{9}$$

where Ss is the user defined sampling scale, ab' is sampled from the distribution φ^i uniformly and randomly, ε' is an added noise vector. The detail of the multi-population based manifold learning algorithm is shown in **Algorithm 4**.

3.4. Procedure of MMOIM

The overall procedure of MMOIM is described in **Algorithm 5**.

Algorithm 5 Procedure of the proposed MMOIM

```
Input: population size $N$, the number of subpopulations $Ns$
Output: $Nondominantpop$
Initialize the population$pop$
while the stopping criterion is not met do
  Divide population $pop$ into $Ns$ subpopulations $subpop_i$, $i = 1, \cdots, Ns$, according to k-means clustering method;
  for each subpopulation $subpop_i$
     Apply the proportional cloning and hypermutation mechanism according to Algorithm 3;
  end for
  Implement the manifold learning strategy by Algorithm 4;
  Sort all the individuals according to the non-dominated sorting and special crowding distance;
Keep the first $N_{store}$ antibodies as the new population and delete the others;
end while
return the non-dominated individuals $Nondominantpop$.
```

A population pop with N antibodies is randomly initialized at first. Then, the multiple population strategy is implemented. Next, the proportional cloning and hypermutation mechanism are applied to each subpopulation according to **Algorithm 3**. After that, the multi-population based manifold learning strategy is applied according to **Algorithm 4**. Then, the antibody population is sorted based on the non-dominated sorting with a special crowding distance method[39] and only the first N_{store} individuals are saved as *Nondominantpop*. The process is repeated until the stopping criterion is met. Finally, the non-dominated individuals *Nondominantpop* are outputted. The main framework of the proposed algorithm is shown in Fig. 2.

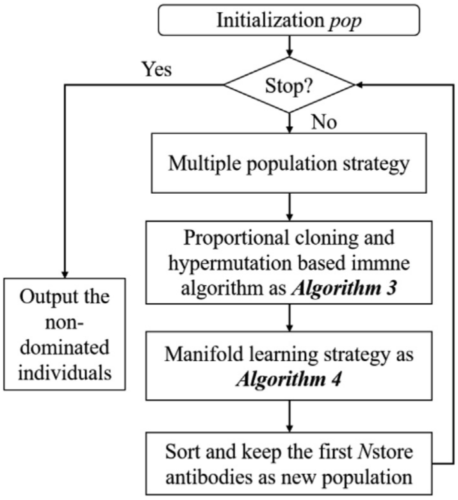

Initialization *pop*
Stop?
Yes
No
Multiple population strategy
Proportional cloning and hypermutation based immne algorithm as ***Algorithm 3***
Manifold learning strategy as ***Algorithm 4***
Sort and keep the first *N*store antibodies as new population
Output the non-dominated individuals

Fig. 2. The Flowchart of the proposed MMOIM.

3.5. Discussions

Three strategies are introduced in the paper, including multi-population mechanism, proportional cloning and hypermutation, and manifold learning strategy.

Firstly, the uniformly distributed population is clustered into subpopulations and each of them covers different searching areas. The multi-population mechanism brings two benefits. The obvious one is maintaining the diverse population due to the parallel and distributed searching process. Moreover, the additional benefit is for preserving the multiple local optima. In MMOPs, there could be more than one Pareto optimal solutions in the decision space corresponding to the same objective value. Due to the distribution of the individuals and the different evolutionary processes, some individuals are supposed to march toward different local optima, and may disappear during the searching process. The main reasons for this are that these individuals are far away in the decision space, and they may map to the same point or are close to each other in the objective space. According to the domination-based selection strategy, once these individuals are dominated, they would not survive. With the multi-population mechanism, these individuals are placed into different subpopulations. Therefore, multiple optimal solutions are approached by different subpopulations. However, multiple subpopulations cover different areas resulting in that some areas may not get explored at all. This problem is addressed by the proportional cloning and manifold learning strategy.

Secondly, the proportional cloning and hypermutation is applied for generating well-distributed solutions. The crowdedness in the objective space in each subpopulation is proportional to its cloning size. More copies are produced with the greater the crowding-distance value. Therefore, there are more searching opportunities in the less crowded areas. If so, the uniformity of the population distribution at the objective frontier is promoted. Additionally, subpopulation size is

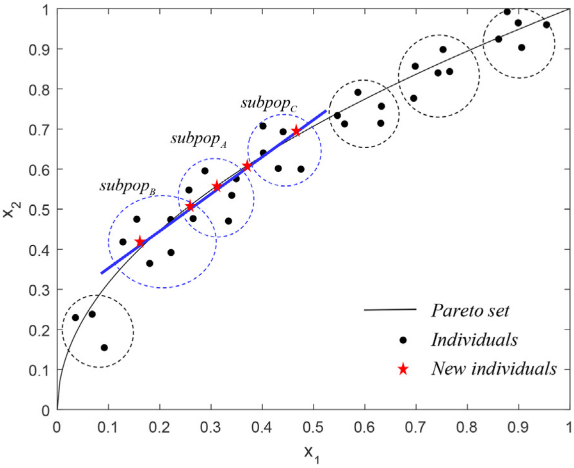


Fig. 3. Illustration of the manifold learning strategy. Manifold learning is performed between the subpopulation composed of individuals scattered around PS and their adjacent subpopulations.

exceeding a predefined clone size, and the harmonized crowding distance is applied in the decision space to select the individuals from the relatively sparse area for cloning. In this case, the uniformity of the population distribution in the decision space is considerably enhanced. Therefore, proportional cloning operator not only benefits for the diversity but also makes the population distribution more uniform.

Thirdly, the manifold learning strategy is applied to promote the convergence toward the Pareto front and the exploration between the subpopulations. As shown in Fig. 3, several individuals belonging to the same subpopulation are surrounded by circles in the decision space. For example, subpopulation $subpop_A$and its two nearest neighbors, $subpop_B$ and $subpop_C$, are selected for building a probability model shown as the blue line according to the manifold learning strategy. Then, the new individuals are sampled from the model shown as the red star. It is observed that the newly generated individuals get closer to the real manifold surface and guide the search to the unexplored area between subpopulations. That is to say, the manifold learning strategy not only plays the role of promoting the local search but also enhance the interaction among subpopulations.

Through the combination of the above strategies, the convergence and diversity of the population are delicately balanced, and the multiple optimal solutions are preserved and distributed uniformly in both decision space and objective space.

3.6. Complexity analysis

When calculating the time complexity, the impact of the decision variables and the objectives are ignored as they are much smaller than the population size. The complexity of k-means clustering is $O(N)$, where N denotes the population size. The complexity of **Algorithm 3** is $O(Ns \cdot CS)$, where Ns and CS refer to subpopulation numbers and the clone scale, respectively. The complexity of **Algorithm 4** is $O(Ns \cdot Ss)$ where Ss the sampling scale. In addition, the complexity of the non-dominated sorting and special crowding distance is $O\left(N + (Ns \cdot CS)^2 + (Ns \cdot Ss)^2\right)$. Overall, since the product of Ns and $CS$is approximately equal to N , and the product of Ns and Ss is also approximately equal to N, the algorithm complexity is approximately $O\left(N^2\right)$, which is comparable with MMO-CLRPSO[10] and SS-MOPSO[37], and less than that of DN-NSGAII345[31], but slightly more than that of MO Ring PSO SCD[30].

4. Experimental studies

To validate the performance of the proposed MMOIM, a standard test suite of instances from 2019 IEEE Congress on Evolutionary Computation, MMOPs competition[39] is used. Four performance matrices, including Pareto Sets Proximity (PSP) [30], Inverted Generational Distance in the decision space (IGDX)[19], Inverted Generational Distance in the objective space (IGDF)[19], and Hypervolume (HV)[40] are adopted to assess the performance of MMOIM both in the decision space and in the objective space. Among them, IGDX indicates the average distance between the obtained PS and the true PSin the decision space, while IGDF evaluates the average distance between the obtained PF and the true PF in the objective space. HV calculates the volume between the obtained PF and a predefined reference point. As a whole, HV and IGDF are proposed as performance metrics for multi-objective optimization, which focuses mainly in the objective space, while IGDX focus on evaluating the performance in the decision space, and PSP would reflect the performance both in the decision space and the objective space simultaneously.

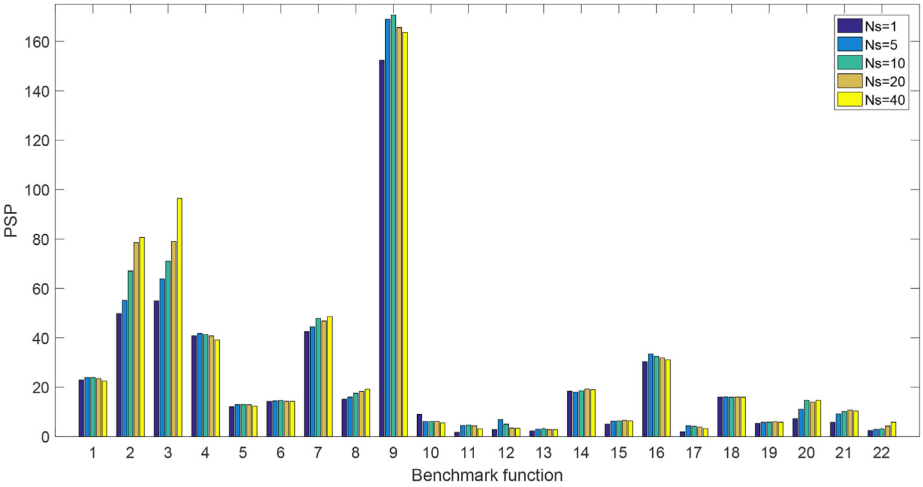


Fig. 4. The mean *PSP* of MMOIM on different clone scale. The value on the horizontal axis indicates the following benchmark function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 16 = MMF1_z, 17 = MMF1_e, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYM-PART rotated, and 22 = Omni-test.

$$HV(\boldsymbol{O}, \boldsymbol{R}) = volume\left(\bigcup_{i=1}^{|\boldsymbol{O}|} \boldsymbol{v}_i\right) \tag{10}$$

where index computes the volume dominated by the set $\boldsymbol{O}$. The larger the value of HV, the better the diversity and convergence of the obtained solution set. Compared with other evaluation indexes, HV does not need the complete true Pareto front beforehend for reference, but only needs one reference point.

$$IGDX(\boldsymbol{O}, \boldsymbol{P}^*) = \frac{\sum_{\boldsymbol{q}\forall \boldsymbol{P}^*} d(\boldsymbol{q}, \boldsymbol{O})}{|\boldsymbol{P}^*|} \tag{11}$$

where $\boldsymbol{P}^*$ indicates a set whose solutions are distributed uniformly in the true PS. All achieved solutions in the decision space form a set denoted as $\boldsymbol{O}$. $d(\boldsymbol{q}, \boldsymbol{O})$ indicates the Euclidean distance between $\boldsymbol{q}$ and a solution in $\boldsymbol{O}$. If the reference point set $\boldsymbol{P}^*$ can well represent the real PS, then IGDX can better measure the diversity and convergence of the solution set in the decision space. The smaller the value of $IGDX$ is, the closer the solution set obtained by the algorithm is to the reference point set. The calculation formula of IGDF is similar to IGDX, where $\boldsymbol{P}^*$ can be expressed by a set formed by the points in the true PF and the solutions in $\boldsymbol{O}$ can be replaced by points distributed in the obtained PF in the objective space.

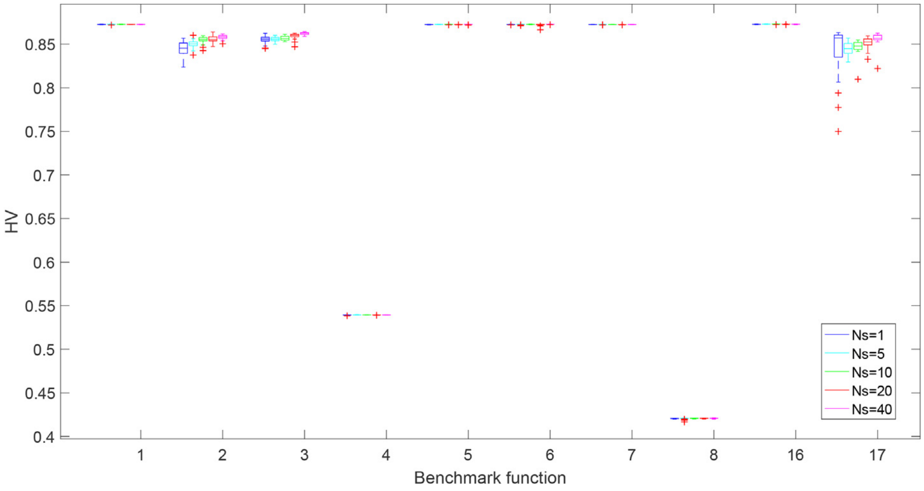


Fig. 5. Comparison on *HV* of MMOIM with different number of subpopulations *Ns*.The value on the horizontal axis indicates the following benchmark function: 1 = MMF1,2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 16 = MMF1_z, and 17 = MMF1_e.

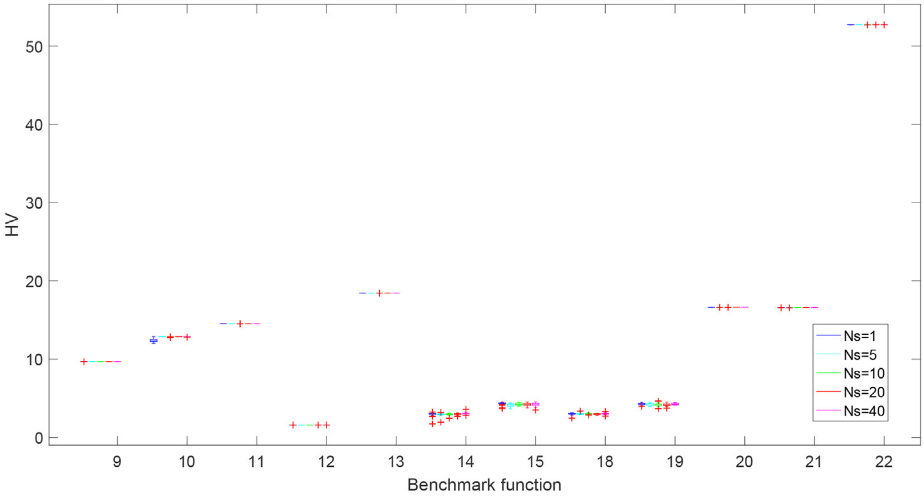


Fig. 6. Comparison on HV of MMOIM with different number of subpopulations *Ns*. The value on the horizontal axis indicates the following benchmark function: 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYM-PART rotated, and 22 = Omni-test.

$$PSP = \frac{CR}{IGDX} \tag{12}$$

where *CR* is the cover rate of the solutions modified from the maximum spread. *CR* can be expressed by

$$CR = \left(\prod_{i=1}^{n} \delta_i \right)^{1/2n} \tag{13}$$

where

$$\delta_i = \begin{cases} 1, & \mathbf{Q}_i^{\max} = \mathbf{Q}_i^{\min} \\ 0, & \boldsymbol{q}_i^{\min} \geqslant \mathbf{Q}_i^{\max} || \boldsymbol{q}_i^{\min} \leqslant \mathbf{Q}_i^{\min} \\ \left(\frac{\min\left(\boldsymbol{q}_i^{\max}, \mathbf{Q}_i^{\max}\right) - \max\left(\boldsymbol{q}_i^{\max}, \mathbf{Q}_i^{\max}\right)}{\mathbf{Q}_i^{\max} - \mathbf{Q}_i^{\min}} \right), & otherwise \end{cases} \tag{14}$$

where $\boldsymbol{q}_i^{\min}$, and $\mathbf{Q}_i^{\max}$ are the minimum of the PS and the true PS in the *i*-dimensional space, respectively, and $\boldsymbol{q}_i^{\max}$ and $\mathbf{Q}_i^{\max}$ are the maximum of the obtained PS and the true PS in the *i*-dimensional space, respectively. The PSP value can faithfully quantify both the convergence of the obtained PS and the degree of similarity between the obtained PS and the true PS.

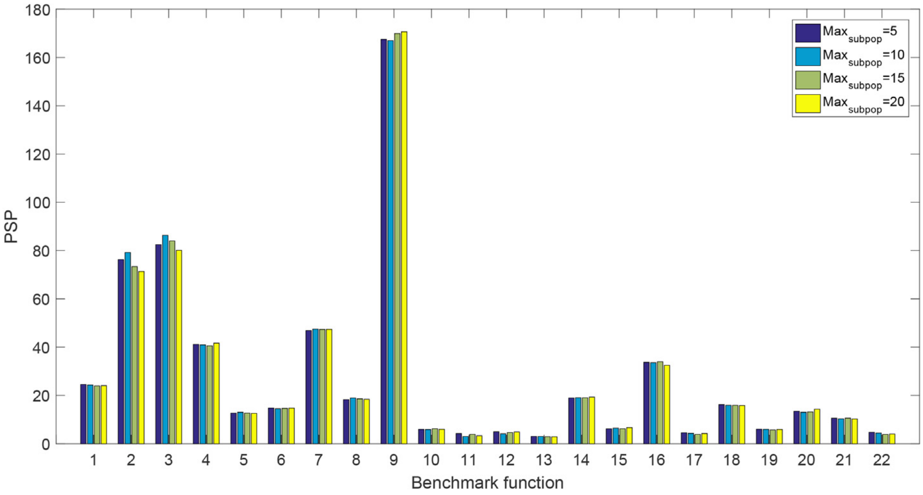


Fig. 7. The mean *PSP* of MMOIM on different Maxsubpop. The value on the horizontal axis indicates the following benchmark function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 16 = MMF1_z, 17 = MMF1_e, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYM-PART rotated, and 22 = Omni-test.

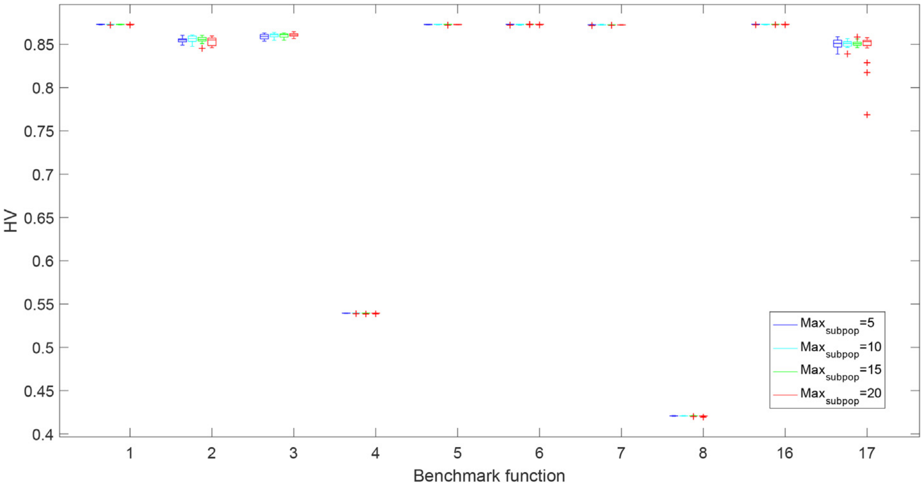


Fig. 8. Comparison on HV of MMOIM on different Maxsubpop. The value on the horizontal axis indicates the following benchmark function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 16 = MMF1_z, and 17 = MMF1_e.

4.1. Experimental setting

To make a fair comparison, the population size N, the maximum fitness evaluations *fitcount* , and the final archive size N_{store} of all algorithms are set to $100 * D$, $5000 * D$ and $100 * D$, respectively, where D is the number of decision variables. All competing algorithms are independently run for 20 times on 22 test problems. The clone scale CS is set to 10 and the mutation probability p_m is set to $1/D$. The number of subpopulations Ns is set to 20. Sampling scale Ss is set to 10, and Maxsubpop is set to 10 by default.

The experiments are implemented with MATLAB 2021a on the 64-bit Microsoft Windows 7 operating system running on a PC with Intel(R) Core(TM) i5-6500 3.20 GHz CPU and 8 GB RAM.

4.2. Discussions of the proposed strategies and introduced parameters

According to the design procedure of the proposed MMOIM, a few tuning parameters are introduced. Most of them came with the operators borrowed from the original sources, such as clonal scale CS, mutation probability p_m , and sampling scale Ss. There are mainly two parameters introduced in MMOIM, including the number of subpopulations Ns and Max_{subpop}. The influence of several strategies and parameters involved in MMOIM is discussed. The *PSP* indicator is shown in a bar chart. To

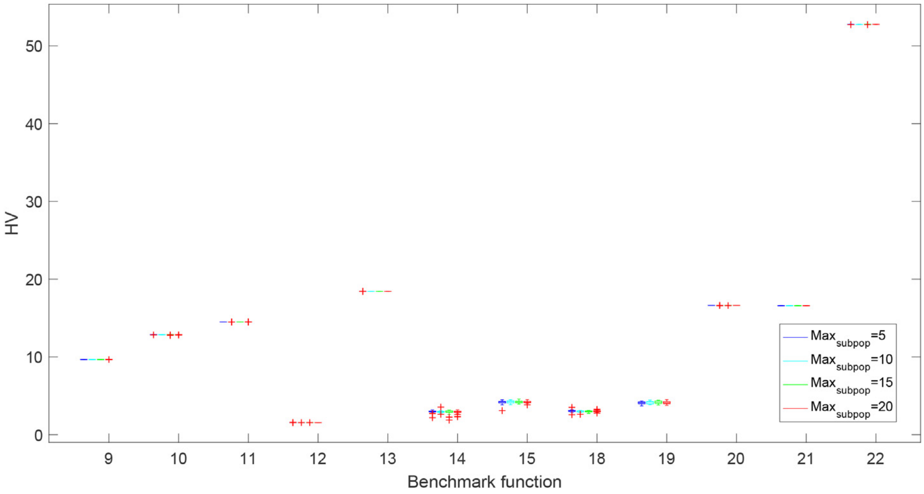


Fig. 9. Comparison on HV of MMOIM on different Maxsubpop. The value on the horizontal axis indicates the following benchmark function: 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYM-PART rotated, and 22 = Omni-test.

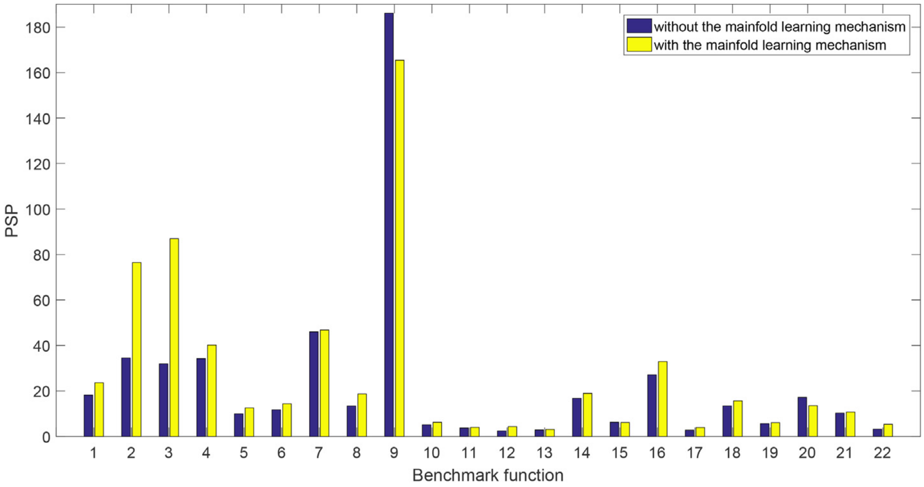


Fig. 10. The mean PSP of MMOIM without and with the multi-population based manifold learning mechanism. The value on the horizontal axis indicates the following benchmark function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 16 = MMF1_z, 17 = MMF1_e, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYM-PART rotated, and 22 = Omni-test.

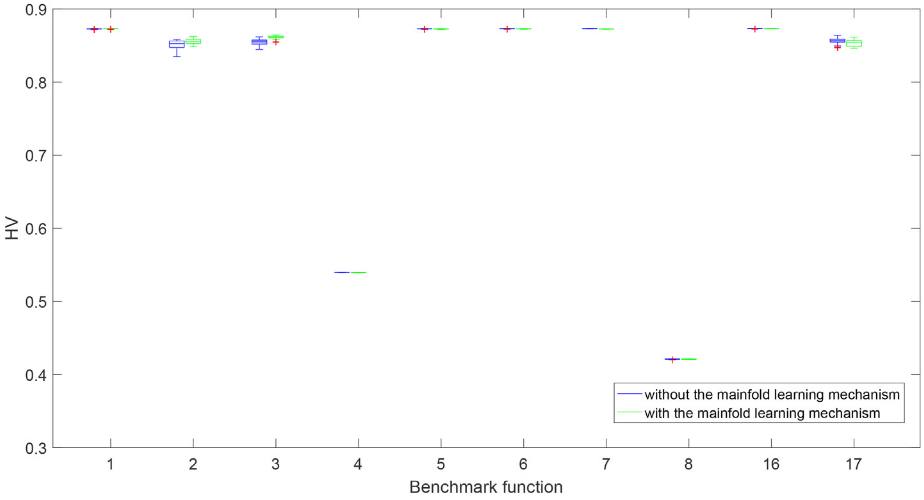


Fig. 11. Comparison on HV of MMOIM without and with the multi-population based manifold learning mechanism. The value on the horizontal axis indicates the following benchmark function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 16 = MMF1_z, and 17 = MMF1_e.

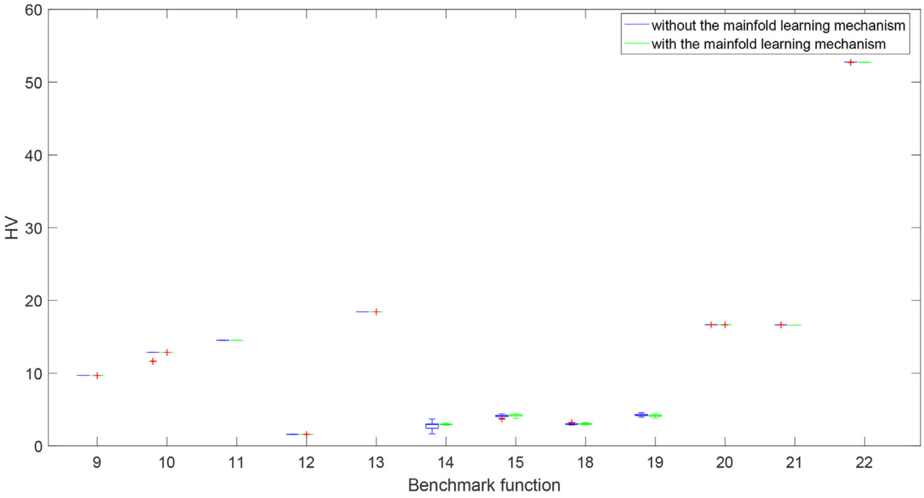
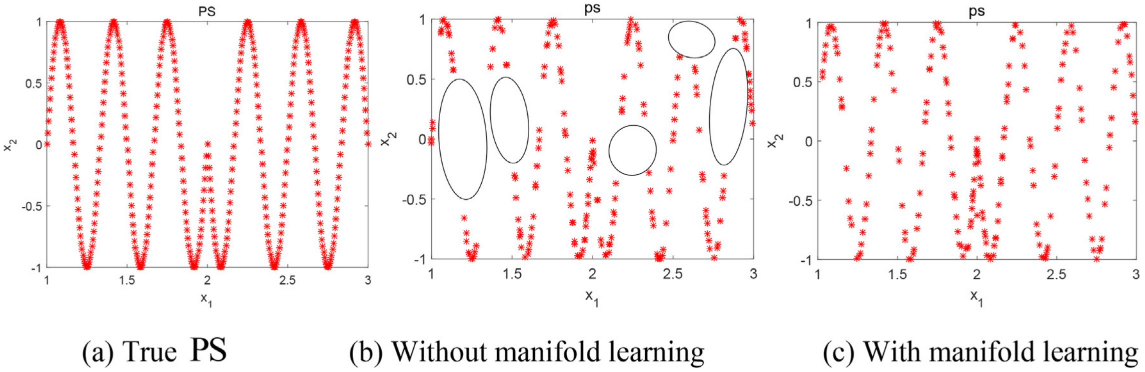


Fig. 12. Comparison on HV of MMOIM without and with the multi-population based manifold learning mechanism. The value on the horizontal axis indicates the following benchmark function: 9 = MMF9, 10 = MMF10, 11 = MMF11, 12 = MMF12, 13 = MMF13, 14 = MMF14, 15 = MMF15, 18 = MMF14_a, 19 = MMF15_a, 20 = SYM-PART simple, 21 = SYMPART rotated, and 22 = Omni-test.



(a) True PS (b) Without manifold learning (c) With manifold learning

Fig. 13. The distribution of the true PS and the obtained PS on MMF1.

make detailed observation, the *HV* value is shown on box diagram. For a better visualization, the results on test functions of *HV* are shown by groups in this section: the MMF1, MMF2, MMF3, MMF4, MMF5, MMF6, MMF7, MMF8, MMF1_z, and MMF1_e are presented in one figure, while the rest are shown in another.

4.2.1. The influence of the number of subpopulationsNs

The parameter *Ns* is introduced into the multi-population strategy to determine subpopulation numbers. The performance of MMOIM with different subpopulation numbers is evaluated. In this experiment, the parameter *Ns* is set to 1, 5, 10, 20, and 40, respectively. The PSP and HV metrics are shown in Figs. 4-6**,** respectively. It is worth noting that all evolutionary operations are performed in a single population when *Ns* is equal to 1. In this case, the multi-population strategy is simply disabled.

It is observed from Fig. 4 that $Ns = 1$ presents the worst performance in comparison to other *Ns* values on most test problems except MMF10. MMOIM with the parameter $Ns = 1$ obtains the best PSP value on MMF10. It is also observed from Figs. 5 and 6 that the same design produces the worse HV value, and it means the proposed MMOIM has not converged sufficiently. That is to say, the good performance in the decision space is at the expense of the performance in the objective space. It is observed that the performance of the proposed MMOIM is improved appreciably with increasing subpopulation numbers on MMF2, MMF3, MMF8, and Omni-test when $Ns > 1$. Therefore, it unambiguously implies that the multi-population strategy

Table 1
The mean value and standard deviation of PSP among competing algorithms.

Function	MMOIM	DN-NSGAII	MO_Ring_PSO_SCD	MMO-CLRPSO	SS-MOPSO	DE-RLFR	MMO-ClusteringPSO
MMF1	23.667	9.663	20.487	23.811	**24.163**	18.958	23.978
	(1.023)	(2.038)+	(0.845)+	(1.275)∼	**(1.272)**∼	(1.246)+	(1.475)-
MMF2	**76.439**	11.646	25.876	27.674	33.164	12.415	27.243
	(11.051)	(6.823)+	(5.256)+	(6.114)+	(9.640)+	(6.242)+	(8.104)+
MMF3	**86.969**	13.040	34.786	35.772	42.104	19.396	35.787
	(10.726)	(3.690)+	(8.126)+	(7.980)+	(7.470)+	(10.851)+	(8.098)+
MMF4	40.217	13.293	36.239	**40.801**	40.218	33.588	38.962
	(2.964)	(2.324)+	(2.086)+	**(2.749)**∼	(2.456)∼	(3.504)+	(4.576)∼
MMF5	12.575	5.856	11.768	13.415	**13.469**	11.315	13.402
	(0.823)	(0.543)+	(0.659)+	(0.821)-	**(0.838)-**	(1.042)+	(0.827)-
MMF6	14.411	6.892	13.754	**15.442**	15.043	13.136	15.044
	(0.911)	(0.642)+	(0.629)+	**(0.850)-**	(0.716)-	(0.989)+	(0.721)-
MMF7	**46.845**	20.006	37.166	40.906	41.066	28.058	40.898
	(3.403)	(4.767)+	(3.080)+	(2.447)+	(3.541)+	(4.629)+	(2.722)+
MMF8	**18.721**	3.948	14.909	16.825	18.469	13.818	17.703
	(1.884)	(1.452)+	(1.540)+	(1.451)+	(1.451)∼	(2.254)+	(1.698)∼
MMF9	**165.453**	53.970	120.360	146.897	144.453	150.939	148.263
	(12.600)	(22.286)+	(8.572)+	(6.937)+	(11.541)+	(33.020)∼	(10.276)+
MMF10	**6.223**	6.080	5.709	5.327	6.110	2.044	5.758
	(0.021)	(3.163)∼	(0.648)+	(1.371)+	(1.318)+	(2.640)+	(0.687)+
MMF11	**3.962**	0.574	3.570	3.928	3.810	0.634	3.370
	(2.024)	(0.050)+	(1.781)+	(1.772)∼	(1.680)∼	(1.155)+	(1.972)+
MMF12	**4.354**	0.482	4.208	3.412	3.576	0.250	3.552
	(2.966)	(0.059)+	(2.621)∼	(2.685)∼	(2.856)∼	(0.061)+	(2.710)∼
MMF13	3.070	1.613	2.946	3.06603	3.019	1.776	**3.162**
	(0.696)	(0.127)+	(0.64)∼	(0.604)∼	(0.762)∼	(0.087)+	**(0.700)**∼
MMF14	**18.958**	10.552	18.684	18.904	17.422		18.352
	(0.502)	(0.533)+	(0.401)∼	(0.427)∼	(0.803)+	+	(0.682)+
MMF15	6.125	3.947	6.530	6.242	**6.586**		6.214
	(0.781)	(0.972)+	(0.710)∼	(0.717)∼	**(0.809)**∼	+	(0.650)∼
MMF1_z	**32.989**	13.749	28.155	32.431	32.525	25.903	32.182
	(1.546)	(2.883)+	(1.272)+	(1.766)∼	(2.275)∼	(2.333)+	(2.353)∼
MMF1_e	**3.925**	0.886	2.004	1.639	2.370	0.372	1.782
	(1.032)	(0.460)+	(0.550)+	(0.467)+	(0.773)+	(0.290)+	(0.680)+
MMF14_a	15.686	8.158	16.349	**16.900**	16.644		16.810
	(0.638)	(0.583)+	(0.602)-	**(0.332)-**	(0.420)-	+	(0.528)-
MMF15_a	6.064	4.420	6.040	6.088	**6.105**		5.877
	(0.508)	(0.666)+	(0.441)∼	(0.500)∼	**(0.397)**∼	+	(0.596)∼
SYM-PARTsimple	13.595	0.257	5.379	5.942	10.530	**13.697**	7.980
	(1.560)	(0.102)+	(0.914)+	(2.127)+	(2.118)+	**(2.216)**∼	(2.582)+
SYM-PARTrotated	**10.707**	0.233	4.889	5.230	10.540	7.263	6.749
	(1.141)	(0.102)+	(1.55)+	(1.851)+	(1.551)∼	(1.363)+	(2.230)+
Omni-test	5.343	0.711	2.737	2.680	3.802	**6.201**	3.163
	(1.709)	(0.141)+	(0.704)+	(0.737)+	(0.936)+	**(2.020)**∼	(0.606)+
sum up all	+ /∼/-	21/1/0	16/5/1	10/9/3	9/10/3	19/3/0	11/7/4

Please note that DE-RLFR cannot converge to the PF on MMF14, MMF14_a, MMF15and MMF15_a.
As a result, no PSPvalue is listed under these test cases for DE-RLFR.

Table 2
The mean value and standard deviation of IGDX among competing algorithms.

Function	MMOIM	DN-NSGAII	MO_Ring_PSO_SCD	MMO-CLRPSO	SS-MOPSO	DE-RLFR	MMO-ClusteringPSO
MMF1	4.208E-02 (1.79E-03)	1.048E-01 (1.74E-02) +	4.847E-02 (1.94E-03) +	4.187E-02 (2.32E-03)∼	**4.129E-02 (2.19E-03)**∼	5.238E-02 (3.46E-03)+	4.165E-02 (2.80E-03)∼
MMF2	**1.328E-02 (2.07E-03)**	1.057E 01 (5.22E-02)+	3.789E 02 (7.28E-03) +	3.602E 02 (8.05E-03) +	3.165E-02 (1.09E-02)+	8.891E-02 (3.65E-02)+	3.907E-02 (1.36E-02)+
MMF3	**1.161E-02 (1.33E-03)**	7.390E 02 (1.85E-02)+	2.922E-02 (7.73E-03) +	2.759E-02 (4.93E-03) +	2.334E-02 (3.61E-03)+	5.746E-02 (2.24E-02)+	2.787E-02 (6.63E-03)+
MMF4	2.491E 02 (1.86E-03)	7.744E 02 (1.50E-02)+	2.741E-02 (1.56E-03) +	**2.447E-02 (1.65E-03)**∼	2.482E-02 (1.61E-03)∼	2.959E-02 (3.28E-03)+	2.593E-02 (3.87E-03)∼
MMF5	7.929E-02 (5.15E-03)	1.695E-01 (1.49E-02) +	8.472E-02 (4.64E-03) +	7.445E-02 (4.85E-03)-	**7.407E-02 (4.75E-03)-**	8.824E-02 (9.24E-03)+	7.443E-02 (4.76E-03)-
MMF6	6.930E-02 (4.49E-03)	1.438E-01 (1.27E-02) +	7.238E-02 (3.37E-03) +	**6.449E-02 (3.39E-03)-**	6.625E-02 (3.09E-03)∼	7.568E-02 (5.49E-03)+	6.618E-02 (3.17E-03)-
MMF7	**2.133E-02 (1.63E-03)**	5.201E-02 (1.32E-02) +	2.680E-02 (2.11E-03) +	2.443E-02 (1.54E-03) +	2.443E-02 (2.19E-03)+	3.443E-02 (5.13E-03)+	2.445E-02 (1.58E-03)+
MMF8	**5.327E-02 (5.06E-03)**	2.710E-01 (8.61E-02) +	6.715E-02 (6.55E-03) +	5.947E-02 (5.16E-03) +	5.402E-02 (4.24E-03)∼	7.372E-02 (1.42E-02)+	5.670E-02 (5.45E-03)+
MMF9	**6.080E-03 (5.00E-04)**	2.210E-02 (9.20E-03) +	8.340E-03 (6.20E-04) +	6.810E-03 (3.20E-04) +	6.970E-03 (6.20E-04)+	7.060E-03 (2.12E-03)∼	6.770E-03 (5.00E-04)+
MMF10	1.604E 01 (1.70E-04)	**1.467E-01 (3.67E-02)**∼	1.683E 01 (1.21E-02)+	1.714E-01 (1.77E-02)+	1.640E-01 (2.47E-02)+	1.902E-01 (1.79E-02)+	1.676E-01 (1.32E-02)+
MMF11	**2.060E-01 (2.69E-02)**	2.505E-01 (2.60E-04) +	2.152E-01 (2.35E-02) +	2.079E-01 (2.42E-02)∼	2.090E-01 (2.31E-02)∼	2.493E-01 (7.63E-03)+	2.139E-01 (2.56E-02)∼
MMF12	**1.852E-01 (4.73E-02)**	2.463E 01 (5.80E-04)+	0.1.878E-01 (4.27E-02)∼	2.000E-01 (4.34E-02)∼	1.979E-01 (4.61E-02)∼	2.467E-01 (2.40E-04)+	1.971E-01 (4.50E-02)∼
MMF13	**2.319E-01 (1.48E-02)**	2.891E-01 (1.18E-02)+	2.378E-01 (1.43E-02)∼	2.374E-01 (1.51E-02)∼	2.366E-01 (1.61E-02)∼	2.600E-01 (5.78E-03)+	2.320E-01 (1.56E-02)∼
MMF14	**5.278E-02 (1.37E-03)**	9.503E-02 (4.84E-03) +	5.354E-02 (1.16E-03)∼	5.292E-02 (1.19E-03)∼	5.752E-02 (2.60E-03)+	+	5.456E-02 (2.05E-03)+
MMF15	1.661E-01 (2.16E-02)	2.368E-01 (2.21E-02)+	1.551E-01 (1.80E-02)∼	1.623E-01 (1.83E-02)∼	**1.543E-01 (1.95E-02)**∼	+	1.628E-01 (1.82E-02)∼
MMF1_z	**3.024E-02 (1.40E-03)**	7.483E-02 (1.67E-02) +	3.532E-02 (1.54E-03) +	3.073E-02 (1.72E-03)∼	3.069E-02 (2.19E-03)∼	3.868E-02 (4.02E-03)+	3.102E-02 (2.33E-03)∼
MMF1_e	**2.568E-01 (6.96E-02)**	1.091E + 00 (5.97E-01) +	4.772E-01 (1.35E-01) +	5.647E-01 (1.24E-01)+	4.252E-01 (1.36E-01)+	1.918E + 00 (6.12E-01)+	5.390E-01 (1.72E-01)+
MMF14_a	6.378E-02 (2.60E-03)	1.232E-01 (8.75E-03) +	6.115E2-0 (2.41E-03)-	**5.906E-02 (1.16E-03)-**	6.004E-02 (1.54E-03)-	+	5.944E-02 (1.85E-03)-
MMF15_a	1.652E 01 (1.31E-02)	2.117E-01 (1.77E-02)+	1.633E 01 (9.64E-03)∼	1.630E 01 (1.18E-02)∼	**1.626E-01 (1.01E-02)**∼	+	1.675E-01 (0.0133)∼
SYM-PARTsimple	**7.445E-02 (8.41E-03)**	4.309E + 00 (1.35E + 00) +	1.913E-01 (3.49E-02) +	2.537E-01 (2.85E-01)+	1.004E-01 (2.78E-02)+	7.489E-02 (1.28E-02)∼	2.056E-01 (2.74E-01)+
SYM-PARTrotated	**9.396E-02 (1.04E-02)**	3.881E + 00 (1.190E + 00) +	2.834E-01 (2.86E-02)+	3.105E-01 (3.52E-02) +	9.664E-02 (1.48E-02)∼	1.409E-1 (2.82E-02)+	2.228E-01 (2.63E-01)+
Omni-test	2.066E-01 (6.40E-02)	1.416E + 00 (2.50E-01)+	3.891E-01 (1.09E-01)+	3.979E-01 (9.95E-02)+	2.755E-01 (5.85E-02)+	**1.877E-01 (9.19E-02)**∼	3.256E-01 (6.28E-02)+
sum up all	+ /∼/-	21/1/0	16/5/1	10/9/3	9/11/2	19/3/0	11/8/3

Please note that DE-RLFR cannot converge to the PF on MMF14, MMF14_a, MMF15and MMF15_a.
As a result, no IGDX value is listed under these test cases for DE-RLFR.

involved in MMOIM is capable of improving the diversity of the population in the decision space. There is almost no obvious regularity on the other benchmark functions with the *Ns* value. As a whole, the recommended *Ns*value is set in the range of $[0.1, 0.2] * N$, where N is the population size.

4.2.2. Discuss the influence of the Max_{subpop}

To study the influence of the parameter Max_{subpop}, a relevant experiment with different values for parameter Max_{subpop} is performed. In this experiment, the parameter Max_{subpop} is set to 5,10, 15, and 20, respectively. The experimental results of *PSP* and *HV* on 22 test problems are shown in Figs. 7-9**,** respectively. It is observed that the proposed MMOIM obtains a promising PSP value on MMF2 and MMF3 when Max_{subpop} is set to 10. MMOIM acquires the best PSP value on MMF9 when Max_{subpop} is set to 20. However, the influence of the parameter Max_{subpop} on *PSP* metric shows no noticeable difference. For *HV*, the algorithms obtain not much difference in performance on all benchmark functions. In short, the proposed MMOIM is not sensitive to the setting of parameter Max_{subpop}.

4.2.3. The effect of the multi-population based manifold learning strategy

In this section, the proposed MMOIM with and without the multi-population based manifold learning strategy is discussed here. The PSP and HV values are shown in Figs. 10-12, respectively. It is observed that the proposed MMOIM algo-

Table 3
The mean value and standard deviation of HV among competing algorithms.

Function	MMOIM	DN-NSGAII	MO_Ring_PSO_SCD	MMO-CLRPSO	SS-MOPSO	DE-RLFR	MMO-ClusteringPSO
MMF1	**8.727E-01** **(2.70E-04)**	8.692E-01 (1.12E-03) +	8.705E-01 (4.00E-04)+	8.714E-01 (4.10E-04)+	8.715E-01 (4.00E-04)+	8.710E-01 (6.30E-04) +	8.713E-01 (5.30E-04)+
MMF2	**8.553E-01** **(3.86E-03)**	8.424E-01 (2.10E-02)∼	8.430E-01 (6.03E-03) +	8.452E-01 (6.09E-03) +	8.490E-01 (6.51E-03) +	8.221E-01 (1.75E-02)+	8.475E-01 (7.51E-03)+
MMF3	**8.611E-01** **(2.45E-03)**	8.430E-01 (2.16E-02) +	8.521E-01 (2.94E-03) +	8.519E-01 (3.91E-03)+	8.550E-01 (2.94E-03)+	8.386E-01 (1.24E-02)+	8.545E-01 (4.20E-03)+
MMF4	**5.394E-01** **(1.50E-04)**	5.383E-01 (3.70E-04) +	5.371E-01 (5.70E-04)+	5.371E-01 (1.10E-03)+	5.374E-01 (8.80E-04)+	5.359E-01 (1.49E-03)+	5.364E-01 (1.21E-03)+
MMF5	**8.725E-01** **(3.00E-04)**	8.694E-01 (4.21E-03) +	8.708E-01 (3.00E-04) +	8.715E-01 (4.20E-04)+	8.715E-01 (3.20E-04)+	8.710E-01 (4.90E-03)+	8.714E-01 (6.90E-04) +
MMF6	**8.726E-01** **(2.60E-04)**	8.701E-01 (1.16E-03) +	8.705E-01 (7.50E-04)+	8.714E-01 (4.80E-04)+	8.715E-01 (4.50E-04)+	8.712E-01 (8.20E-04)+	8.712E-01 (5.80E-04)+
MMF7	**8.725E-01** **(2.00E-04)**	8.683E-01 (3.94E-03) +	8.707E-01 (4.30E-04) +	8.705E-01 (9.00E-04)+	8.700E-01 (8.90E-04)+	8.707E-01 (1.44E-03)+	8.701E-01 (9.50E-04)+
MMF8	**4.208E-01** **(2.20E-04)**	4.197E-01 (5.90E-04)+	4.142E-01 (4.15E-03) +	4.162E-01 (3.65E-03) +	4.188E-01 (1.65E-03) +	4.093E-01 (2.58E-02)+	4.180E-01 (2.52E-03)+
MMF9	**9.678E + 00** **(6.58E + 00)**	9.676E + 00 (2.68E-03) +	9.670E + 00 (3.91E-03) +	9.663E + 00 (6.17E-03) +	9.655E + 00 (2.27E-02)+	9.669E + 00 (1.41E-02)+	9.664E + 00 (9.57E-03)+
MMF10	**1.285E + 01** **(6.99E-03)**	1.214E + 01 (3.91E-01) +	1.253E + 01 (1.04E-01)+	1.254E + 01 (7.38E-02)+	1.259E + 01 (7.62E-02) +	1.268E + 01 (4.11E-01)∼	1.256E + 01 (1.07E-01)+
MMF11	**1.451E + 01** **(2.84E-03)**	1.450E + 01 (2.11E-03) +	1.449E + 01 (4.08E-03) +	1.450E + 01 (3.54E-03)+	1.450E + 01 (8.09E-03)+	1.449E + 01 (2.11E-02)+	1.450E + 01 (6.72E-03) +
MMF12	**1.572E + 00** **(9.80E-04)**	1.572E + 00 (8.10E-04)+	1.564E + 00 (3.35E-03) +	1.563E + 00 (6.59E-03) +	1.565E + 00 (3.23E-03)+	1.567E + 00 (6.24E-03) +	1.564E + 00 (4.67E-03) +
MMF13	**1.844E + 01** **(1.05E-03)**	1.843E + 01 (6.59E-03) +	1.838E + 01 (1.14E-02)+	1.839E + 01 (1.34E-02)+	1.840E + 01 (8.18E-03)+	1.842E + 01 (3.56E-03) +	1.841E + 01 (8.95E-03)+
MMF14	2.970E + 00 (8.98E-02)	**3.050E + 00** **(6.63E-02)-**	2.840E + 00 (1.65E-01) +	2.920E + 00 (2.89E-01)∼	2.830E + 00 (1.24E-01)+	+	2.810E + 00 (2.21E-01)+
MMF15	4.183E + 00 (1.67E-01)	**4.341E + 00** **(1.76E-01)**	4.165E + 00 (1.42E-01)∼	4.194E + 00 (1.48E-01)∼	4.299E + 00 (1.78E-01)∼	+	4.191E + 00 (1.84E-01)∼
MMF1_z	**8.729E-01** **(1.60E-04)**	8.705E-01 (9.30E-04) +	8.707E-01 (4.90E-04)+	8.714E-01 (5.00E-04)+	8.715E-01 (3.20E-04)+	8.713E-01 (5.10E-04)+	8.710E-01 (3.70E-04)+
MMF1_e	8.534E-01 (4.69E-03)	7.474E-01 (1.85E-01) +	8.286E-01 (2.87E-02)+	8.481E-01 (1.03E-02)∼	8.520E-01 (1.04E-02)∼	**8.649E-01** **(9.30E-03)-**	8.513E-01 (1.32E-02)∼
MMF14_a	3.042E + 00 (9.51E-02)	**3.169E + 00** **(1.18E-01)-**	3.018E + 00 (1.45E-01)∼	2.960E + 00 (2.18E-01)∼	2.975E + 00 (1.25E-01)∼	+	3.022E + 00 (2.91E-01)∼
MMF15_a	4.129E + 00 (1.71E-01)	**4.323E + 00** **(2.45E-01)-**	4.190E + 00 (1.27E-01)∼	4.286E + 00 (1.51E-01)-	4.271E + 00 (1.27E-01)-	+	4.285E + 00 (1.30E-01)-
SYM-PARTsimple	1.663E + 01 (3.54E-03)	**1.664E + 01** **(3.10E-03)-**	1.653E + 01 (2.19E-02)+	1.656E + 01 (1.67E-02) +	1.663E + 01 (4.90E-03)+	1.664E + 01 (3.90E-03)-	1.659E + 01 (8.80E-03)+
SYM-PARTrotated	1.660E + 01 (9.38E-03)	**1.663E + 01** **(2.83E-03)-**	1.650E + 01 (2.01E-02)+	1.655E + 01 (1.39E-02)+	1.662E + 01 (6.11E-03)-	1.661E + 01 (9.83E-03)∼	1.658E + 01 (1.35E-02)+
Omni-test	5.273E + 01 (1.88E-02)	**5.279E + 01** **(3.67E-03)-**	5.258E + 01 (4.44E-02)+	5.262E + 01 (3.32E-02)+	5.270E + 01 (2.15E-02)+	5.269E + 01 (4.17E-02)+	5.264E + 01 (3.15E-02)+
sum up all	+ /∽/-	14/1/7	19/3/0	17/4/1	17/3/2	18/2/2	18/3/1

Please note that DE-RLFR cannot converge to the PF on MMF14, MMF14_a, MMF15and MMF15_a.
As a result, no HV value is listed under these test cases for DE-RLFR.

rithm with multi-population based manifold learning strategy obtains the best PSP values on 15 out of 22 test problems. There is no significant difference on MMF11, MMF13, MMF15, MMF15_a, and SYM-PARTrotated. And, the proposed MMOIM algorithm without multi-population based manifold learning strategy achieves a better performance only on MMF9 and SYM-PARTsimple. There is no appreciable difference on HV values. It suggests that the multi-population based manifold learning strategy is beneficial for improving the performance in the decision space and maintaining the performance in the objective space, simultaneously. Based on this case, it is concluded that the manifold learning strategy is helpful for improving the performance of the proposed MMOIM algorithm on most of the benchmark functions.

To further analyze the results, the distribution of the true PSand the obtained PSby the proposed algorithm, MMOIM, with and without the multi-population based manifold learning strategy on MMF1 is displayed. The distribution of the true PS, obtained PS without and with manifold learning on MMF1 are shown in Fig. 13**(a-c)** respectively. It is observed that these PS obtained by MMOIM without manifold learning strategy cannot cover the whole PS, and these PS are marked as the black circle in Fig. 13**(b)**. Combined with the manifold learning strategy, the distribution of the obtained PS is more uniform as displayed in Fig. 13**(c)**. It suggests that the multi-population based manifold learning strategy guides the population to explore the unexplored area. However, the manifold leaning mechanism needs to consume extra computational resources. If the manifold learning mechanism could not learn the real structure of the true PS, the mechanism may be useless or even bring negative influence to the final results given the limitation of the fitness evaluations.

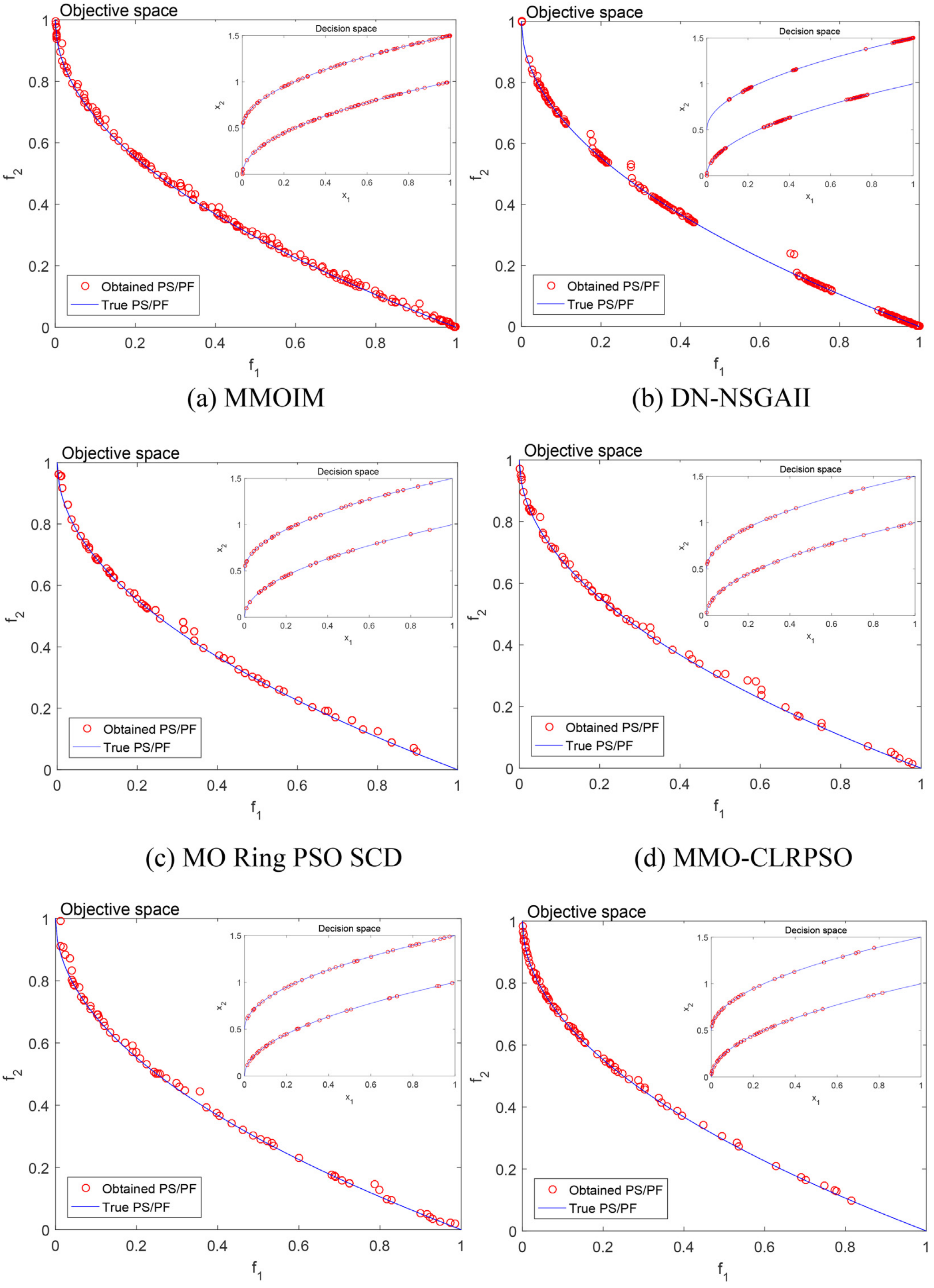
Table 4
The mean value and standard deviation of IGDF among competing algorithms.

Function	MMOIM	DN-NSGAII	MO_Ring_PSO_SCD	MMO-CLRPSO	SS-MOPSO	DE-RLFR	MMO-ClusteringPSO
MMF1	**2.610E-03**	4.930E-03	3.790E-03	3.160E-03	3.150E-03+	3.660E-03	3.270E-03
	(1.20E-04)	(9.40E-04) +	(1.90E-04)+	(2.20E-04)+	(2.30E-04)+	(2.80E-04)+	(2.40E-04)+
MMF2	**8.610E-03**	2.447E-02	2.096E-02	1.917E-02	1.758E-02	4.576E-02	1.755E-02
	(8.20E-04)	(1.63E-02) +	(3.71E-03)+	(4.49E-03)+	(8.43E-03)+	(2.06E-02)+	(4.48E-03)+
MMF3	**7.190E-03**	2.345E-02	1.522E-02	1.505E-02	1.223E-02	2.971E-02	1.338E-02
	(6.70E-04)	(1.52E-02) +	(2.50E-03)+	(2.72E-03)+	(1.40E-03)+	(1.17E-02)+	(2.86E-03)+
MMF4	**2.600E-03**	3.050E-03	3.630E-03	3.630E-03	3.500E-03	3.760E-03	4.080E-03
	(2.10E-04)	(2.30E-04) +	(3.30E-04)+	(6.80E-04)+	(4.70E-04)+	(3.60E-04)+	(7.80E-04)+
MMF5	**2.620E-03**	3.960E-03	3.650E-03	3.060E-03	3.130E-03	3.650E-03	3.190E-03
	(7.00E-05)	(7.30E-04) +	(1.50E-04)+	(1.20E-04)+	(1.60E-04)+	(3.20E-04)+	(3.20E-04)+
MMF6	**2.530E-03**	3.780E-03	3.550E-03	3.050E-03	3.10E-03	3.450E-03	3.20E-03
	(1.00E-04)	(2.90E-04) +	(1.80E-04)+	(1.20E-04)+	(2.20E-04)+	(3.40E-04)+	(2.50E-04)+
MMF7	**2.680E-03**	4.020E-03	3.810E-03	3.820E-03	4.140E-03	3.870E-03	4.160E-03
	(8.00E-05)	(2.70E-04)+	(2.80E-04)+	(5.00E-04)+	(5.00E-04)+	(4.20E-04)+	(5.10E-04)+
MMF8	**3.20E-03**	4.110E-03	4.940E-03	3.810E-03	3.390E-03	3.970E-03	3.550E-03
	(2.20E-04)	(5.30E-04) +	(6.00E-04) +	(2.70E-04)+	(2.20E-04)+	(3.80E-04)+	(2.10E-04)+
MMF9	**1.407E-02**	1.440E-02	1.594E-02	1.921E-02	2.342E-02	1.888E-02	1.931E-02
	(3.27E-03	(1.26E-03) +	(1.75E-03)+	(2.77E-03)+	(1.00E-02)+	(3.91E-03)+	(4.06E-03)+
MMF10	**1.503E-01**	1.879E-01	2.055E-01	2.025E-01	1.944E-01	1.815E-01	2.016E-01
	(8.39E-03)	(4.72E-02)+	(2.06E-02) +	(2.25E-02)+	(1.25E-02)+	(1.97E-02)+	(2.03E-02)+
MMF11	**8.284E-02**	9.771E-02	8.714E-02	8.393E-02	8.891E-02	1.075E-01	8.879E-02
	(4.43E-03)	(1.18E-03) +	(6.67E-03) +	(5.97E-03)∼	(6.88E-03)+	(8.73E-03)+	(6.68E-03)+
MMF12	**5.973E-02**	8.337E-02	6.086E-02	6.492E-02	6.156E-02	8.862E-02	6.810E-02
	(1.46E-02)	(4.40E-04)+	(1.05E-02)∼	(1.19E-02)+	(1.37E-02)∼	(3.14E-03)+	(1.35E-02)+
MMF13	**7.743E-02**	1.530E-01	9.836E-02	8.384E-02	9.558E-02	1.438E-01	9.329E-02
	(2.00E-02)	(7.26E-03) +	(1.747E-02)+	(1.10E-02)+	(2.19E-02)+	(1.83E-02)+	(2.27E-02)+
MMF14	**7.571E-02**	1.111E-01	7.998E-02	7.998E-02	8.868E-02		8.072E-02
	(2.03E-03)	(7.79E-03) +	(2.03E-03)+	(2.60E-03)+	(4.31E-03)+	+	(2.41E-03)+
MMF15	1.722E-01	2.129E-01	1.732E-01	**1.722E-01**	1.767E-01		1.746E-01
	(6.20E-03)	(9.40E-03) +	(3.36E-03)∼	**(4.53E-03)**∼	(5.48E-03)+	+	(4.82E-03)∼
MMF1_z	**2.520E-03**	3.740E-03	3.610E-03	3.060E-03	3.090E-03	3.380E-03	3.200E-03
	(1.00E-04)	(4.70E-04)+	(1.40E-04)+	(1.40E-04)+	(1.50E-04)+	(2.70E-04)+	(1.90E-04)+
MMF1_e	8.450E-03	2.693E-02	1.208E-02	1.272E-02	9.560E-03	**8.350E-03**	1.059E-02
	(1.25E-03)	(1.59E-02) +	(1.85E-03)+	(1.76E-03)+	(1.47E-03)+	**(8.21E-03)-**	(1.86E-03)+
MMF14_a	**7.496E-02**	1.232E-01	7.772E-02	7.593E-02	7.800E-02		7.657E-02
	(1.40E-03)	(7.15E-03) +	(2.15E-03)+	(1.62E-03)+	(2.15E-03)+	+	(1.85E-03)+
MMF15_a	1.759E-01	2.259E-01	1.737E-01	1.736E-01	**1.750E-01**		1.746E-01
	(3.87E-03)	(9.75E-03) +	(2.09E-03)-	(3.33E-03)-	**(3.01E-03)**∼	+	(4.13E-03)∼
SYM-PARTsimple	1.470E-02	**1.306E-02**	4.020E-02	3.280E-02	1.622E-02	2.025E-02	2.472E-02
	(2.01E-03)	**(1.86E-03)-**	(5.80E-03)+	(5.56E-03) +	(2.41E-03)+	(5.44E-03)+	(3.58E-03)+
SYM-PARTrotated	1.862E-02	**1.491E-02**	4.686E-02	3.511E-02	1.820E-02	2.675E-02	2.988E-02
	(2.11E-03)	**(2.87E-03)-**	(7.70E-03)+	(5.50E-03) +	(1.99E-03)∼	(5.22E-03)+	(4.13E-03)+
Omni-test	1.185E-02	**8.020E-03**	4.283E-02	3.375E-02	2.321E-02	1.999E-02	2.944E-02
	(7.60E-04)	**(5.20E-04)-**	(7.45E-03)+	(4.28E-03)+	(2.39E-03)+	(2.07E-03)+	(3.78E-03)+
sum up all	+ /∼/-	19/0/3	19/2/1	19/2/1	19/3/0	21/0/1	20/2/0

Please note that DE-RLFR cannot converge to the PF on MMF14, MMF14_a, MMF15and MMF15_a.
As a result, no IGDF value is listed under these test cases for DE-RLFR.

4.3. Comparison with other state-of-the-art algorithms

To demonstrate the effectiveness of MMOIM, the MMOIM is compared with six state-of-the-art algorithms, including DN-NSGAII[31], MO_Ring_PSO_ SCD[30], MMO-CLRPSO[11], SS-MOPSO[37], DE-RLFR[41] and MMO-Clustering PSO[42]. The PSP,



(a) MMOIM

(b) DN-NSGAII

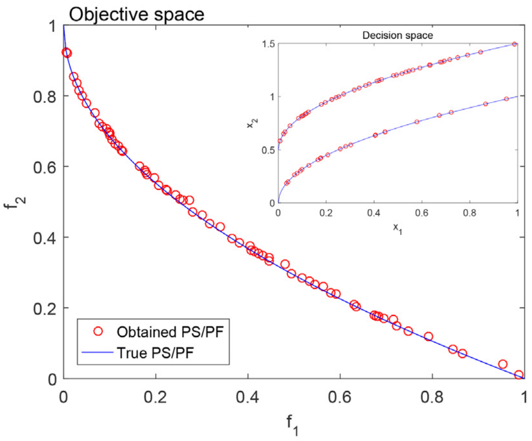
(c) MO Ring PSO SCD

(d) MMO-CLRPSO

(e) SS-MOPSO

(f) DE-RLFR

Fig. 14. The distribution of solutions obtained by seven competing algorithms on MMF3.



(g) MMO-Clustering PSO

Fig. 14 (*continued*)

IGDX, HV, and IGDF values of MMOIM and six competing algorithms are listed in Tables. 1-4, respectively. In addition, the Wilcoxon's rank sum test is used to measure the significance of performance between MMOIM and one of its competing algorithms at the significant level of 0.05. The symbols '+', '-' and '∼' indicate that the proposed MMOIM is statistically better than, worse than, or similar to the chosen competing algorithm. For each benchmark function, the best performance of all algorithms is highlighted in bold.

4.3.1. Comparisons in the decision space

As shown in Table. 1, the mean value and standard deviation of PSP values by the six compared algorithms are summarized. MMOIM obtains the best PSP value on 12 out of 22 problems. MMOIM performs significantly better than the other algorithms on MMF2 and MMF3. Although SS-MOPSO acquires the best PSP value on MMF1, MMF5, MMF15, and MMF15_a, its overall performance is still worse than that of MMOIM. MMO-CLRPSO and MO_Ring_PSO_SCD obtain slightly inferior performance. Although MMO-CLRPSO performs the best on three test problems (MMF4, MMF6, and MMF14_a), it still performed worse than MMOIM on ten of the test functions. Especially, MO_Ring_PSO_SCD is inferior to MMOIM on all of the test functions. For SYM-PART simple and Omni-test, DE-RLFR shows the superiority over the compared algorithm, however it is unapplicable to implement on MMF14, MMF14_a, MMF15, and MMF15_a. Although MMO-Clustering PSO acquires the best PSP on MMF13, MMOIM also gets good values, while just a little bit worse. On the other hand, DN-NSGAII exhibits the worst performance on most functions.

IGDX is another indicator commonly used for evaluating the distribution of the obtained Pareto optimal solutions in the decision space. The compared results of IGDX are shown in Table. 2. Obviously, the IGDX values are similar to those of PSP, but there are some differences. MMOIM still obtains the best performance on most of the test functions. From the analysis of the experimental results, the ranking order of the next one is SS-MOPSO, as it has an excellent performance on four test functions. DE-RLFR still shows good performance on Omni-test, but has trouble in performing on the other functions. Both MMO-Clustering PSO and MO_Ring_PSO_SCD perform moderately on all of the test functions. For DN-NSGAII, both IGDX values and PSP values show similar performance, which clearly implies that DN-NSGAII remains to be improved in solving MMOPs.

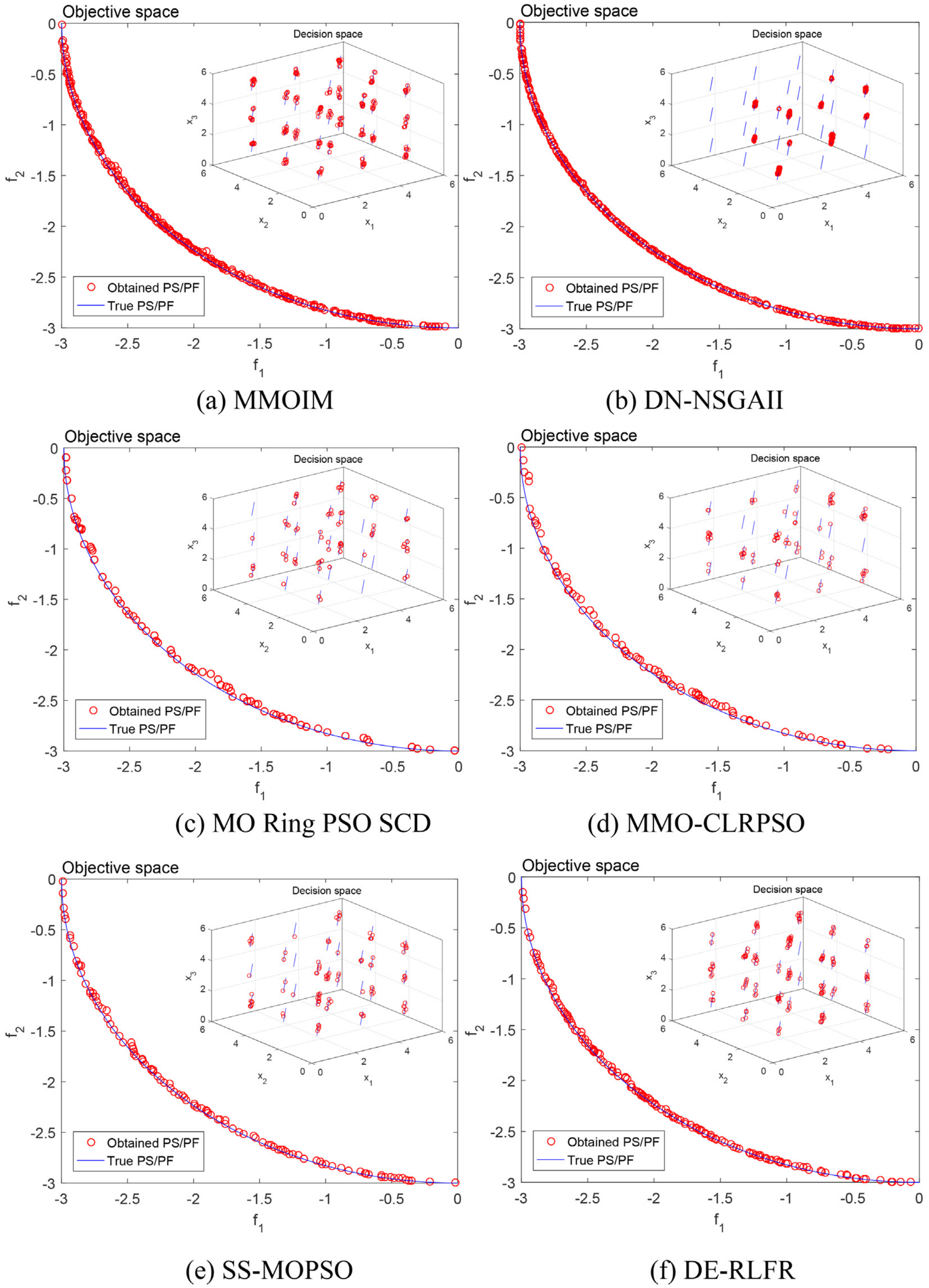
In summary, it is concluded that the proposed MMOIM is competitive in comparison to the chosen state-of-the-art algorithms in the decision space.

4.3.2. Comparisons in the objective space

HV values of the proposed algorithm and the other compared algorithms are reported in Table. 3. Generally, the pursuit of greater distribution in the decision space is likely to deteriorate their performance in the objective space. Luckily, it can be observed that MMOIM not only achieves better performance in the decision space in most cases, but also performs extremely well in the objective space. DN-NSGAII which ranks second only to MMOIM in terms of HV achieves the best values on seven test problems. As to the other four algorithms, their performance is much worse than the two algorithms described above. Although DE-RLFR achieves the optimal performance on MMF1_e, it is unsatisfied in the decision space according to the evaluations of PSP and IGDX. The other four algorithms are far inferior to the above algorithms.

The comparisons of IGDF among six compared algorithms are also discussed and displayed in Table. 4. Intuitively, the proposed MMOIM achieves excellent performance on 16 out of 22 functions.

Although DN-NSGAII obtains the best IGDF on SYM-PART simple, SYM-PART rotated, and Omni-test, its performance in the decision space is obviously worse than other algorithms. The IGDF obtained by DE- RLFR is similar to HV. Besides, MMO-CLRPSO achieves the best mean IGDF values on MMF15 and MMF15_a. As for MO_Ring_PSO_SCD, MMO-Clustering PSO, and SS-MOPSO, they are less competitive in the objective space. Based on the above analysis of extensive experimental results, the proposed algorithm, MMOIM, not only obtains well-distributed PSs, but also could approach to the true PF closely.



(a) MMOIM

(b) DN-NSGAII

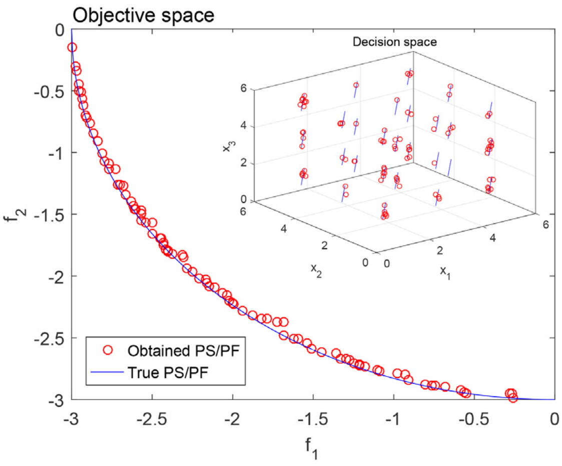
(c) MO Ring PSO SCD

(d) MMO-CLRPSO

(e) SS-MOPSO

(f) DE-RLFR

Fig. 15. The distribution of solutions obtained by seven competing algorithms on Omni-test.



(g) MMO-Clustering PSO

Fig. 15 (*continued*)

4.4. Performance analysis in both the decision and objective spaces

To discuss the performance of the competing algorithms further, the distribution of Pareto optimal solutions obtained by all competing algorithms on two representative functions (MMF3 and Omni-test) in both decision space and objective space is analyzed. The distribution of solutions obtained by seven competing algorithms is shown in Figs. 14-15 respectively, where the red asterisk represents the obtained PSs in the decision space. It is worthy of noting that the out-of-ranged solutions are discarded for better visual comparisons in the same axes ranges.

MMF3 is a typical MMOP that has two equivalent Pareto optimal solutions in the decision space with the same objective values in the objective space. It is visually observed from Fig. 14 that DN-NSGAII performs the worst since both the obtainedPS and the PFhave multiple large fractures against the real PS and PF. Even though there are scattered and discrete points, the rest of the compared algorithms perform much better in both decision and objective spaces. Especially, both obtained PS and obtained PF by the proposed MMOIM are more uniform and comprehensive than its competing algorithms.

It is worth noting that Omni-test has 27 equivalent PSs with the same PF. As we know, multimodal multi-objective optimization problems with multiple PSs require more computational resources resulting in higher difficulty. As shown in Fig. 15, the obtained solutions by MMOIM can almost cover the true PSs on Omni-test. Also, DE-RLFR has a similarly excellent performance in the decision space. The performance of DN-NSGAII cannot locate multiple equivalent Pareto optimal solutions, and it is weaker than other competing algorithms on Omni-test. The other five competing algorithms are able to locate more PSs. When turning to the objective space, the obtained PF by most of these competing algorithms is capable of converging to the true PF. Especially, MMOIM and DN-NSGAII are superior to other algorithms. For simplicity, MMOIM achieves satisfactory performance in both objective space and decision space.

5. Conclusions and future work

In this paper, we have proposed a cluster-based immune-inspired algorithm using manifold learning for multimodal multi-objective optimization. The population is divided into multiple subpopulations by clustering, and multiple subpopulations and proportional cloning in the immune-inspired algorithm are designed to locate equivalent PSs in the decision space. Additionally, manifold learning is incorporated into the proposed MMOIM and presents a benefit for convergence. The seamless combination of the three strategies achieves a promising balance between convergence and diversity in bothe the decision and objective spaces. The experimental results show that the proposed MMOIM is competitive in both decision space and objective space in comparison to some chosen competing algorithms.

On the other hand, clustering, cloning, and manifold learning operators consume a vast amount of computational resources. How to appreciably reduce the complexity of the algorithm remains our future works. Moreover, manifold learning has a great influence on generating candidate solutions. Therefore, the exploration to approximating the true PS is a valuable research direction as well.

CRediT authorship contribution statement

Weiwei Zhang: This author is responsible for the proposed evolutionary methodology development and experimental validation. **Ningjun Zhang:** This author is involved in data curation and analysis. **Weizheng Zhang:** This author assists in

the experimental validation. **Gary G. Yen:** This author is responsible for the research conceptualization. **Guoqing Li:** This author assisted in responsible for project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the Key research and development and promotion special project of Henan province (NO. 212102210154, 192102110203), Science and technology research key project of basic research projects in education department of Henan province (No. 20A520004) and Youth core teacher training program of universities in Henan province (NO. 2019GGJS138).

References

[1] H. Zhang, Q. Zhang, L. Ma, Z. Zhang, Y. Liu, A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows, Inf. Sci. 490 (2019) 166–190.

[2] J.N. Kuk, R.A. Gonçalves, L.M. Pavelski, S.M. Guse Scós Venske, C.P. de Almeida, A.T. Ramirez Pozo, An Empirical Analysis of Constraint Handling on Evolutionary Multi-objective Algorithms for the Environmental/Economic Load Dispatch Problem, Expert Syst. Appl. 165 (2021) 113774, https://doi.org/10.1016/j.eswa.2020.113774.

[3] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, Integr. Comput.-Aided Eng. 24 (4) (2017) 315–335.

[4] C. Yue, J. Liang, B. Qu, et al, A novel multiobjective optimization algorithm for sparse signal reconstruction, Signal Process. 167 (2020) 107292.

[5] G. Li, W. Wang, W. Zhang, et al, Handling multimodal multi-objective problems through sel-organizing quantum-inspired particle swarm optimization, Inf. Sci. 577 (2021) 510–540.

[6] Y. Liu, G.G. Yen, D. Gong, A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies, IEEE Trans. Evol. Comput. 23 (4) (2019) 660–674.

[7] Y. Hu, J. Wang, J. Liang, K. Yu, H. Song, Q. Guo, C. Yue, Y. Wang, A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm, Sci. China Informat. Sci. 62 (7) (2019) 1–17.

[8] R. Tanabe, H. Ishibuchi, A niching indicator-based multi-modal many-objective optimizer, Swarm Evol. Comput. 49 (2019) 134–146.

[9] R. Xu, H. Wang, S. Zhu, H. Jiang, Z. Li, Multiobjective planning for spacecraft reorientation under complex pointing constraints, Aerosp. Sci. Technol. 104 (2020) 106002, https://doi.org/10.1016/j.ast.2020.106002.

[10] J. Ma, J. Zhang, Y. Lin, Z. Dai, Cost-efficiency trade-offs of the human brain network revealed by a multiobjective evolutionary algorithm, NeuroImage 236 (2021) 118040, https://doi.org/10.1016/j.neuroimage.2021.118040.

[11] W. Zhang, G. Li, W. Zhang, J. Liang, G.G. Yen, A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization, Swarm Evol. Comput. 50 (2019) 100569.

[12] K. Zhang, C. Shen, J. He, G.G. Yen, Knee based multimodal multi-objective evolutionary algorithm for decision making, Inf. Sci. 544 (3) (2020) 39–55.

[13] W. Sheng, X. Wang, Z. Wang, Q. Li, Y. Chen, Adaptive Memetic Differential Evolution with Niching Competition and Supporting Archive Strategies for Multimodal Optimization, Inf. Sci. 573 (2021) 316–331.

[14] W. Hu, G.G. Yen, G. Luo, Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system, IEEE Trans. Cybern. 47 (6) (2017) 1446–1459.

[15] G. Li, W. Wang, W. Zhang, W. You, F. Wu, H. Tu, Handling multimodal multi-objective problems through self-organizing quantum-inspired particle swarm optimization, Inf. Sci. 577 (2021) 510–540.

[16] C. Hu, H. Ishibuchi, Incorporation of a decision space diversity maintenance mechanism into MOEA/D for multi-modal multi-objective optimization, in: in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2018, pp. 1898–1901.

[17] L. Weerasena, M.M. Wiecek, B. Soylu, An algorithm for approximating the pareto set of the multiobjective set covering problem, Ann. Oper. Res. 248 (1-2) (2017) 493–514.

[18] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63.

[19] A. Zhou, Q. Zhang, Y. Jin, Approximating the set of pareto-optimal solutions in both the decision and ob jective spaces by an estimation of distribution algorithm, IEEE Trans. Evol. Comput. 13 (5) (2009) 1167–1189.

[20] M. Gong, L. Jiao, H. Du, L. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evol. Comput. 16 (2) (2008) 225–255.

[21] X. Liu, M. Shan, R. Zhang, L. Zhang, Green vehicle routing optimization based on carbon emission and multiobjective hybrid quantum immune algorithm, Mathemat. Probl. Eng. 2018 (2018) 1–9.

[22] Z. Huang, Y. Zhou, Runtime analysis of immune-inspired hypermutation operators in evolutionary multi-objective optimization, Swarm Evol. Comput. 65 (2021) 100934, https://doi.org/10.1016/j.swevo.2021.100934.

[23] F. Caraffini, F. Neri, M. Epitropakis, Hyperspam: A study on hyper-heuristic coordination strategies in the continuous domain, Inf. Sci. 477 (2019) 186–202.

[24] F. Neri, S. Rostami, Generalised pattern search based on covariance matrix diagonalisation, SN Computer Science 2 (3) (2021) 1–22.

[25] F. Neri "Adaptive Covariance Pattern Search", EvoApplications 2021, Lecture Notes in Computer Science, Vol 12694.

[26] F. Neri, S. Rostami, A local search for numerical optimisation based on covariance matrix diagonalisation 12104 (2020) 3-19.

[27] R. Tanabe, H. Ishibuchi, A review of evolutionary multimodal multiob jective optimization, IEEE Trans. Evol. Comput. 24 (1) (2020) 193–200.

[28] H. Xia, J. Zhuang, D. Yu, Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization, IEEE transactions on cybernetics 44, (3) (2013) 378-393.

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

[30] C. Yue, B. Qu, J. Liang, A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems, IEEE Trans. Evol. Comput. 22 (5) (2018) 805–817.

[31] J.J. Liang, C.T. Yue, B.Y. Qu, Multimodal multi-objective optimization: A preliminary study, 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE 2016 (2016) 2454–2461.

[32] K. Zhang, M. Chen, X. Xu, G.G. Yen, Multi-objective evolution strategy for multimodal multi-objective optimization, Appl. Soft Comput. J. 101 (2020) 107004.

[33] R. Tanabe, H. Ishibuchi, A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, 2018, pp. 249-261.

[34] K. Maity, R. Sengupta, S. Saha, Multimodal neighborhood-sensitive archived evolutionary many-objective optimization algorithm, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE 2019 (2019) 286–294.

[35] R. Shi, W. Lin, Q. Lin, Z. Zhu, J. Chen, Multimodal multi-objective optimization using a density-based one-by-one update strategy, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE 2019 (2019) 295–301.

[36] J. Liang, Q. Guo, C. Yue, et al, A sel-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems, in: International Conference on Swarm Intelligence, Springer, 2018, pp. 550-560.

[37] B. Qu, C. Li, J. Liang, et al, A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems, Appl. Soft Comput. 86 (2020) 105886.

[38] R. Sengupta, M. Pal, S. Saha, S. Bandyopadhyay, NAEMO: Neighborhood-sensitive archived evolutionary many-objective optimization algorithm, Swarm Evol. Comput. 46 (2019) 201–218.

[39] C. Yue, B. Qu, K. Yu, J. Liang, X. Li, A novel scalable test problem suite for multimodal multiobjective optimization, Swarm Evol. Comput. 48 (2019) 62–71.

[40] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimisation problems, Swarm Evol. Comput. 34 (2017) 50–67.

[41] Z. Li, L. Shi, C. Yue, Z. Shang, B. Qu, Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems, Swarm Evol. Comput. 49 (2019) 234–244.

[42] W. Zhang, G. Li, W. Zhang, Decision variable clustering based PSO for multimodal multiobjective optimization, 2019 IEEE Congress on Evolutionary Computation (CEC), 2019.