Swarm and Evolutionary Computation 50 (2019) 100569

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization

Weizheng Zhang, Guoqing Li, Weiwei Zhang, Jing Liang, Gary G. Yen *

ARTICLE INFO

Keywords:
Particle swarm optimization
Clustering
Ring topology
Multimodal multi-objective optimization

ABSTRACT

In the multimodal multi-objective optimization problems (MMOPs), there exists more than one Pareto optimal solutions in the decision space corresponding to the same location on the Pareto front in the objective space. To solve the MMOPs, the designed algorithm is supposed to converge to the accurate and well-distributed Pareto front, and at the same time to search for the multiple Pareto optimal solutions in the decision space. This paper presents a new cluster based particle swarm optimization algorithm (PSO) with leader updating mechanism and ring-topology for solving MMOPs. Multiple subpopulations are formed by a new decision variable clustering method with the aim of searching for the multiple Pareto optima solutions and maintaining the diversity. Global-best PSO is employed for independent evolution of subpopulations, while local-best PSO with ring topology is used to enhance the information interaction among subpopulations. Seamlessly integrated, the proposed algorithm provides a good balance between exploration and exploitation. In addition, leader updating strategy is introduced to identify the best leaders in PSO. The performance of the proposed algorithm is compared with six state-of-the-art designs over 11 multimodal multi-objective optimization test functions. Experimental results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

It is not unusual that two or more conflicting objectives are involving in many practical optimization problems. For example, in manufacturing industry, spare parts purchasing is a crucial task to manufacturing firms. Price and quality are commonly considered when purchasing the spare parts. Finding the adequate spare parts with good quality and low price is one of the main considered issues of decision makers. However, better quality usually implies higher price, i.e., price and quality are two conflict objectives. This kind of problem is called multi-objective optimization problems (MOPs). Generally, the multi-objective optimization problem can be defined as:

$$\min_{X} \quad F(X) = (f_1(X), \cdots, f_m(X)) \\ \text{subject to } X \in \Omega \tag{1}$$

where $X = (x_1, \cdots, x_n)$ is an n dimensional decision vector in Ω, where $\Omega \subset R^n$ is the feasible *decision space*, $F: R^n \to R^m$ is the m dimensional objective vector, and all the possible values of objective vectors compose the *objective space*. In view of the objectives in MOPs are often conflicting with one another, the comparison of different solutions is based on the Pareto dominance relationship: a solution X is said to dominate another solution X', if $\forall i \in \{1, \cdots, m\}$, $f_i(X) \leq f_i(X')$ and $\exists i \in \{1, \cdots, m\}$, $f_i(X) < f_i(X')$. The set containing all the non-dominated solutions in the decision space is called as *Pareto optimal Set* (PS), while its mapping in the objective space is referred to as *Pareto optimal Front* (PF).

Over the last two decades, plenty of multi-objective evolutionary algorithms (MOEAs) have been proposed to solve MOPs [1–5]. Recently, the multi-objective optimization problems which constitute two or more distinct PSs corresponding to the same PF have drawn much attentions. This kind of problems is termed “multimodal multi-objective optimization problems (MMOPs)” [6]. Continuing on the spare parts example, different suppliers could provide spare parts with different 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 objectives as shown in the right subfigure of Fig. 1. 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 as shown in the left subfigure in Fig. 1. That is to say, to an optimal solution in the PF (an optimal spare parts as shown in highlighted dot), there are more than one solutions in the PS (multiple suppliers could produce the spare parts with the same price and quality by using different materials or techniques as shown in the left of Fig. 1).

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

https://doi.org/10.1016/j.swevo.2019.100569
Received 23 January 2019; Received in revised form 1 June 2019; Accepted 22 August 2019
Available online 26 August 2019
2210-6502/

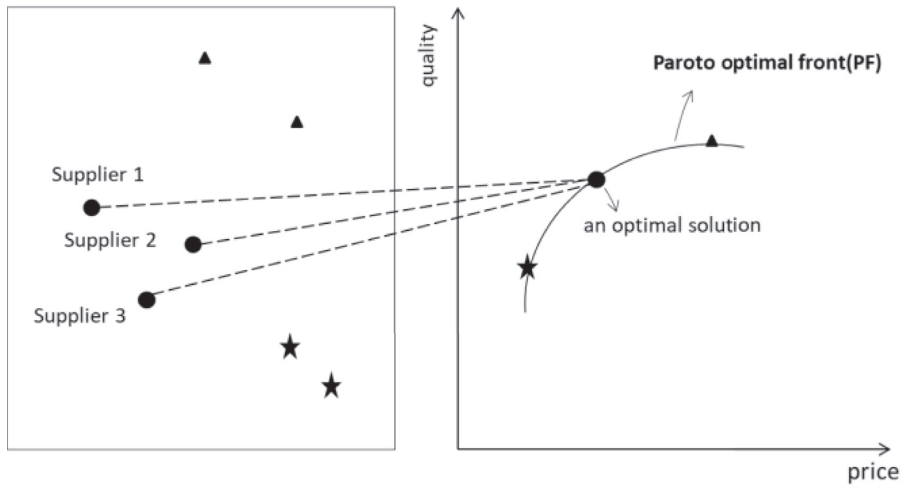


Fig. 1. Example of multimodal multi-objective optimization through spare parts purchasing.

In this example, the optimal spare parts construct the PF in the objective space, while all the suppliers who could provide these spare parts compose the PS in the decision space. If a specific supplier suffers from the spare parts shortage due to the misjudgment of the market or other unexpected reasons, known other suppliers who could provide the spare parts with the same quality and price would be available to maintain the normal production. Therefore, it is meaningful to not only find the PF in the objective space but also to locate the multiple PSs in the decision space.

Multimodal is one of the basic characteristics for both single-objective and multi-objective optimization problems. Especially, there are plenty of works done in solving the multimodal single-objective optimization problems [7–10]. To locate the multiple local optima instead of one global optimum, fitness sharing [11,12], crowding [13,14], clearing [15], speciation [16] and other niching methods were presented. However, little attention has been paid to multimodal multi-objective optimization. Moreover, even though most multi-objective optimization problems are inherited with multimodality, most researches focus on searching for the PF in the objective space, instead of locating multiple PSs in the decision space. Zhou [17] addressed that the distribution of the PS in the decision space is worthy to study and presented two typical classes of problems. 1) A finite number of different PSs may map to the same PF. 2) The dimensionality of the PS may be higher than that of PF. The class 1 scenario refers to the multimodal multi-objective optimization problems, however, the proposed algorithm in Ref. [17] remains focus on the class 2 problems. It is found that it is important to maintain the diversity in the decision space when handling the multi-objective optimization problems [18], and accordingly the crowding distance in the decision space is studied [19,20], and the clustering [21] and other niche techniques were proposed to approximate the PS. Therefore, a decision variable clustering method was proposed in this paper to manipulate the distribution of the solutions in the decision space through partition of the population into subpopulations. Each particle of the subpopulations evolves independently.

Due to the effectiveness and robustness concerns, particle swarm optimization (PSO) method has been widely applied in both academic and engineering community in the last two decades [22–24]. For practical purpose, many PSO variants have been proposed. The velocity and position of each particle are adjusted dynamically according to its personal best position *Pbest*, the best position achieved so far by the whole population, *Gbest* (in the global version PSO), or the best position found so far in its neighborhood, *Nbest* (in the local version PSO). These leaders (including *Pbest*, *Gbest* and *Nbest*) guiding mechanisms are found to be effective. It is straightforward that the leader should be updated if new particles with higher fitness values are found. However, things get complicated in multi-objective optimization problems. The leader should be updated by the new particle if it is dominated. However, they often suffer from the situation that newly generated particles neither dominate nor be dominated by the current leaders when addressing the multi-objective optimization problems. Extra archives are maintained to deposit the mutually non-dominating particles and the leaders are selected from the archives in some previous works [25,26]. It is obvious that the extra archives will increase the space and time complexities. In addition to the extra archive, there are three designs available: 1) replace the leader by the mutually non-dominating particle [27]; 2) according to some rule, e.g., a certain probability [28]; 3) do not update the leaders until they are dominated. In multimodal multi-objective optimization, updating the leaders by the mutually non-dominating particle as presented in case one will bring the frequent replacement of the leaders and hereby affect the searching efficiency. In addition, an extra parameter will be introduced and the performance is somewhat poor in case two. Therefore, leader updating mechanism case three is adopted in this paper.

It is known that population topology could enhance the population diversity when solving the multimodal or multi-objective optimization problems [29]. Ring topology is one of the most commonly used topology when handling the multimodal optimization problems since it could be used as a niching algorithm without requiring defining niching parameters [30]. Through connected by the index, the particles could interact with its neighbors in the ring. Inspired by this, local-best PSO using ring topology is implemented on the non-dominated solutions found in each subpopulation to enhance the exploration in this paper.

This work proposes a novel algorithm to solve the multimodal multi-objective optimization problems, in which the decision variable clustering method, novel leader updating mechanism, and local-best PSO using ring topology are seamlessly integrated. The major contributions of this paper are the following. 1) A new clustering method based on the Euclidean distance in the decision space is proposed. The decision variable clustering method is designed to divide the population into multiple uniformly distributed subpopulations in the same size, which are expected to locate and converge to the multiple PSs independently. 2) Global-best PSO and local-best PSO are incorporated together for accelerating the convergence to the true PFs and at the same time maintain the diversity in the decision space. The global-best PSO is employed for the evolution of subpopulations, while the local-best PSO connects the subpopulations end to end by ring topology to reinforce the exploration and avoid stagnation. Moreover, the ring topology could promote the information interaction among subpopulations. 3) The leader updating mechanism in PSO is introduced to identify the best leaders, in which a leader is updated only when it is dominated. The new leader updating strategy is instrumental in improving the convergence and avoiding the frequent replacements.

The rest of this paper is organized as follows. In Section 2, we review the existing works related to PSO, non-dominated sorting and crowding distance, and the available solutions to MMOPs in recent years. In Section 3, we introduce the proposed algorithm in detail. In Section 4, we conduct experimental simulations and comparison. The effects of involving mechanisms and design parameters are analyzed as well. Finally, conclusion is drawn in Section 5 providing relevant observations and future research directions.

2. Related work

2.1. Particle swarm optimization

Particle Swarm Optimization (PSO) is a population-based algorithm inspired by the flocking behavior of birds [31]. The potential solutions are considered as particles flying in the search space of the problem domain. Their velocities and positions are adjusted according to their best position achieved so far, the global best position found by the population, or possible the best position achieved in its neighborhood so far. The most known design is called as the global-best PSO and the updating of velocity and position of each particle is according to Equations (2) and (3),

$$V_i^{'} = wV_i + c_1r_1(Pbest_i - X_i) + c_2r_2(Gbest - X_i) \tag{2}$$

$$X_i^{'} = X_i + V_i \tag{3}$$

where V_i and $V_i^{'}$ denote the current and updated velocities of particle i, X_i and $X_i^{'}$ denote the current and updated positions of particle i, $Pbest_i$ denotes the personal best position of particle i, $Gbest$ refers to the global optimal position, r_1 and r_2 are random numbers with a normal distribution between zero and one, and w, c_1 and c_2 are tuning constants.

Another design is the local-best PSO in which the best position $Gbest$ is replaced by $Nbest$ which is the best positon achieved so for within its neighborhood. The velocity update of the local PSO is given below:

$$V_i^{'} = wV_i + c_1r_1(Pbest_i - X_i) + c_2r_2(Nbest_i - X_i) \tag{4}$$

where $Nbest_i$ denotes the historical best position of the neighborhood for particle i. Generally, the neighborhood is determined by the adopted population topology. A comprehensive review has been carried out to summarize the involving population topologies for PSO and DE [29]. Among population topology models, the ring topology is widely adopted in handling the multimodal or multi-objectives optimization problems [30]. In the ring topology, the individuals are connected in a ring and each individual only interacts with its left and right neighbors. In Ref. [29], it shows several population structures in PSO to improve exploration ability, such as ring/lbest structure, ring/gbest structure, wheels, random, etc. As suggested by Li [30], PSO with ring topology could be used as a niching algorithm to locate multiple optima without requiring niching parameters.

The population is divided into multiple subpopulations and each subpopulation is isolated from each other and evolved independently and simultaneously in the distributed model [32]. Clustering [33] is one representative way to partition the population. In Ref. [34], the clustering was employed to divide the population into multiple subpopulations so that different subpopulations can locate multiple solutions. In Ref. [35], Yang introduced a novel density clustering based niching method to locate and track global and local optima for a multimodal function. Through independent evolution of multiple subpopulations, the exploitation could be enhanced.

2.2. Non-dominated sorting and crowding distance

Comparing with the single-objective optimization problems, it is hard to choose an exact solution as the optimal solution for a given multi-objective optimization problem. Goldberg [36] put forward the Pareto dominance to sort out a set of solution for MOPs. Over the past decades, some non-dominated sorting algorithms have been developed to improve the efficiency of non-dominated sorting, including order ranking [37], grid dominance [38], relaxed form of Pareto dominance [39], etc. The dominant comparisons between solutions are the main operation for non-dominant sorting. Most existing non-dominated sorting methods consider reducing the number of comparisons to improve computational efficiency. However, the non-dominated sorting only focuses on distribution of solutions in the objective space.

To estimate the density of solutions in the objective space, Deb proposed the crowing distance in NSGA-II [40]. The crowding distance of the ith solution $CD_{i,obj}$, was determined by the average side length of the cuboid formed by the nearest neighbors in the objective space. In Omni-optimizer [41], crowding distance was extended into the decision variable space, denoted as $CD_{i,\mathrm{var}}$. A max or min step is applied on both the objective space and the decision variable space crowding distances as shown in Equation (5) to determine the final crowding distance.

$$CD_i = \begin{cases} \max\left(CD_{i,\mathrm{var}}, CD_{i,obj}\right) & CD_{i,\mathrm{var}} > CD_{avg,\mathrm{var}} \quad or \quad CD_{i,obj} > CD_{avg,obj} \\ \min\left(CD_{i,\mathrm{var}}, CD_{i,obj}\right) & otherwise \end{cases} \tag{5}$$

where $CD_{avg,\mathrm{var}}$ and $CD_{avg,obj}$ are the average crowing distances in the decision and objective spaces, respectively. Based on Deb's work, Yue [42] developed a special crowding distance to multimodal multi-objective optimization, in which the boundary points in the objective space are reassigned to 0 or 1 (according to the objective value) instead of infinity as in the previous works. Under the calculation, a solution whose neighbors are far away from it in both spaces will acquire a large crowing distance, i.e., the methodology promotes diversity of Pareto optimal solutions in both the decision space and the objective space simultaneously.

2.3. Previous works on multimodal multi-objective optimization

Most multi-objective optimization algorithms pay attentions to the distribution of PF in the objective space rather than the distribution of PS in the decision space. Some works have been done on manipulating the distribution of the solutions in the decision space. Schutze et al. [43] proposed an epsilon-domination technique to handle the problems in which two solutions are similar (not equivalent) in the objective space but different in the decision space. Deb [41] first introduced the concept of crowding distance in the decision space and pointed out that maintaining good distribution is not equivalent to locating more Pareto optimal solutions. Xia [18] proposed a multi-objective evolutionary algorithm (MOEA) framework including two crowding estimation methods, multiple selection methods and search strategies for multi-objective evolutionary optimization, in which two crowding distances were designed to maintain the diversity both in the decision space and the objective space. Zhou [17] analyzed the scenario that a good approximation to the PF might not approximate the PS well and proposed a probabilistic model based on MOEA to handle the situation that dimensionalities of the PS and the PF manifolds are different. These works considered the importance of PS in the decision space. However, the situation where multiple PSs mapping with the same PF when handling multi-objective optimization problems is still not well studied.

Until recently, Liang [6] defined the situation as the multimodal multi-objective optimization problems (MMOPs), and proposed an algorithm named DN-NSGAII to locate the multiple PSs. Based on the previous work, Yue et al. [42] made a significant progress in the MMOPs field, in which the real-world applications were illustrated at first, and then a set of multimodal multi-objective optimization benchmark functions along with an evaluation indicator was designed, and finally a novel particle swarm optimizer with the redesigned crowding distance was proposed. After that, Liang [44] proposed the SMPSO-MMO algorithm for MMOPs, which used a self-organizing mechanism to build a population structure to avoid premature convergence. Based on SMPSO-MMO, Hu [45] proposed the MMOPIO algorithm, which employs the self-organizing map with a modified pigeon-inspired optimization and special crowding distance to locate multiple Pareto optimal solutions in the decision space. Additionally, Liu [46] proposed a novel multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies to guarantee diversity in the objective space and promote the diversity in the decision space. Liang [47] proposed a multimodal multi-objective Differential Evolution optimization algorithm, in which a novel decision-variable preselection scheme was introduced for promoting the diversity of solutions in both the decision and the objective spaces.

It can be found that the involving research in the MMOPs field is relatively few. The principle and method of handling MMOPs is not well studied. An effective method to both obtain the PF in the objective space and locate the multiple PSs in the decision space is worthy to pursue. The effectiveness of the strategy in handling multimodal and multi-objective optimization problems needs to be developed. Therefore, in the paper, we propose a cluster based PSO with leader updating mechanism and ring-topology (in short for MMO-CLRPSO) for Multimodal Multi-objective Optimization.

3. Proposed algorithm: MMO-CLRPSO

In the section, we present the main framework of the proposed cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization (MMO-CLRPSO), which is composed of three components: decision variable clustering method, global-best PSO for subpopulation evolution, and local-best PSO with ring topology for non-dominated solutions search. The details are shown in the following subsections.

3.1. Main framework of the proposed algorithm

Algorithm 1
The framework of proposed MMO-CLRPSO

Input: N (population size), Num (number of clusters)
1: Initialize the population *pop*;
2: Divide the population into Num subpopulations $\{\text{subpop}_1, \cdots, \text{subpop}_{\text{Num}}\}$ according to decision variable clustering method;
3: For each subpopulation $subpop_k$
4: Sort the subpopulation according to the non-dominated sorting and special crowding distance;
5: Save the non-dominated solutions into non-dominated set $NondomiSet_k$;
6: Set first particle in non-dominated set $NondomiSet_k$ as the global best $subGbest_k$;
7: EndFor
8: While termination criterion not fulfilled **do**
9: Global-best PSO for subpopulation evolution;
10: Ring topology based PSO for local search;
11: End While
12: Output the non-dominated particles in every non-dominated set.

Algorithm 1 presents the main framework of the proposed MMO-CLRPSO. A population *pop* with N particles is randomly initialized at first. Then, the population is clustered into Num subpopulations $\{\text{subpop}_1, \cdots, subpop_{Num}\}$, where Num denotes the number of clusters, according to the proposed decision variable clustering method in the decision space (line 2). Next, the particles in each subpopulation are sorted based on the non-dominated sorting and special crowding distance method, and then the non-dominated solutions of each subpopulation are saved into non-dominated set $\{\text{nondomiSet}_k\}$ $k = 1, \cdots, Num$, and the first particles in the sorted subpopulations are marked as global best of each subpopulation $\{\text{subGbest}_k\}$, $k = 1, \cdots, Num$. Next, each subpopulation starts to evolve based on the global-best PSO, and then the ring topology based PSO is implemented on the non-dominated set for additional local search. The process is repeated until a termination condition is met. Finally, the non-dominated particles in every non-dominated set are considered as outputs.

3.2. Decision variable clustering method

There are multiple optimal solutions in the decision space corresponding to one optimal solution in the objective space in a given multimodal multi-objective optimization problem. Multi-population method which is able to locate multiple optima simultaneously is considered as an effective way to handle the multimodality. Many clustering methods [48,49] could be used.

Algorithm 2
Decision Variable Clustering (pop, Ns)

Initialize Num empty subpopulation set $subpop_k = \varphi$, $k = 1, \cdots, Num$;
$k = 1$;
while $size(pop) > 0$
  if $size(subpop_k) == 0$
    $subpop_k\{1\} \leftarrow pop\{1\}$ //Put the first particle of the *pop*into kth subpopulation;
    delete the first particle $pop\{1\}$ from the population *pop*;
    $size(subpop_k) = size(subpop_k) + 1$;
    $size(pop) = size(pop) - 1$;
  else if $size(subpop_k) < Ns$
    for $m = 1 : size(pop)$;
      for$n = 1 : size(subpop_k)$
        $dis(m, n) \leftarrow$ Euclidean distance between $pop\{m\}$ and $subpop_k\{n\}$
      endfor
      $mdis(m) = \sum_{n=1}^{size(subpop_k)} dis(m, n)/size(subpop_k)$;
    endfor
    Find the particle with minimal *mdis*, put the particle into $subpop_k$ and delete it from *pop*
    $size(subpop_k) = size(subpop_k) + 1$;
    $size(pop) = size(pop) - 1$;
  else
    Put $subpop_k$ into *subpopSet*;
    $k = k + 1$;
  endif
endwhile
output *subpopSet*.

In this paper, a simple decision variable clustering method is proposed to divide the population into multiple subpopulations uniformly based on the Euclidean distances among particles. The clustering process starts with Num empty sets to deposit the subpopulations and hereby each of the subpopulation should have $Ns = N/Num$ particles. In order to arrive at the uniform clustering, one dimension with maximum standard deviation of particles in population *pop* is chosen and denoted as *ChooseDim*. The particles in population *pop* are sorted in the ascending order according to the chosen dimension *ChooseDim* at first. The first particle in the *pop* is placed into the first subpopulation $subpop_1$ and removed from *pop*. Then the average Euclidean distance between each particle in the *pop* and the particles in the subpopulation are calculated. If the size of the subpopulation is less than Ns, the particle with minimal average Euclidean distance will be added until the size of the subpopulation reaches Ns. Otherwise, the first particle in the *pop* will be placed into a new subpopulation. Delete the underlying particle from *pop*. The iteration goes on until *pop* is empty. In this way, particles in the *pop* will be distributed into the corresponding subpopulations set $subpopSet = \{\text{subpop}_1, \cdots subpop_{Num}\}$. The decision variable clustering method is presented in Algorithm 2.

Fig. 2 presents an example to illustrate the proposed clustering method. The process starts with population *pop* with N particles and the number of cluster is Num. The particle in the first index is put into the first

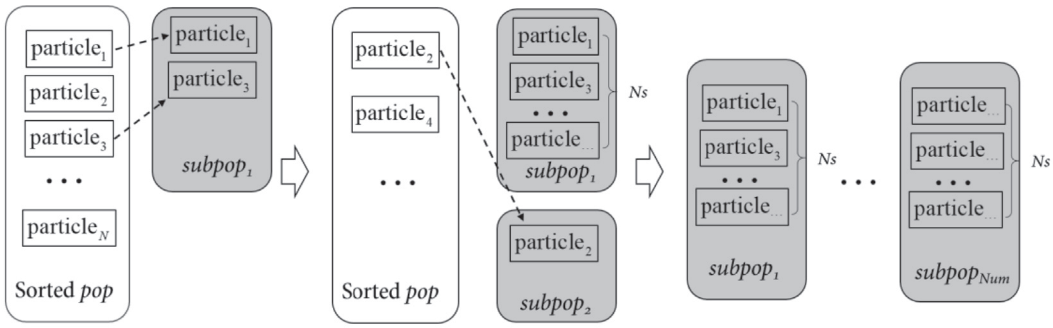


Fig. 2. Illustration of the proposed clustering method.

subpopulation $subpop_1$ and gets removed from *pop*. Afterward, the average Euclidean distance between each particle in the *pop* and all particles in the subpopulation $subpop_1$ is calculated. The particle with the smallest average Euclidean distance will be added into $subpop_1$ and removed from *pop* as well. The process will repeat till the size of $subpop_1$ reaches *Ns*. Then the top particle in the list of the *pop* will be added into the second subpopulation $subpop_2$ and get removed from *pop*. The process will repeat until *pop* is empty.

From the above description, it can be observed that *Num* subpopulations with equal size will be created automatically in the decision space by using the above clustering method. Here we take an example to illustrate the reason why *pop* is sorted according to the chosen dimension *ChooseDim* at first. It is assumed that there are 15 particles in the *pop*, which is to be clustered into three subpopulations as displayed in Fig. 3(a). According to the clustering process, it is obvious that different order of *pop* leads to different clustering results since the first particle of each subpopulation is taken from *pop* in order. Three possible scenarios are shown in Fig. 3(b)-(d), respectively. Fig. 3(b) represents a situation with random sorted *pop*, the individual labelled with the cross is the first chosen particle to subpopulation $subpop_1$, and then four particles are added into the same subpopulation according to the Euclidean distance. Then the individual labelled with the triangle is selected as the first member of subgroup $subpop_2$ and four particles are added in the same way. Finally, subpopulation $subpop_3$ is formed with the individual labelled with the pentagram as the first member. It is observed that the distribution of the three subpopulations is not uniform. In Fig. 3(c) and (d), the particles sorted according to x_1–dimension and x_2–dimension, respectively at first. From Fig. 3(a), it is observed that the distribution of particles is more dispersive in x_1–dimension than x_2–dimension if the same coordinate scale is assumed. Accordingly, the subpopulations in Fig. 3(c) distribute more evenly than those in Fig. 3(d) after clustering. Therefore, x_2–dimension which has the maximum standard deviation is chosen to sort the particles.

3.3. Global-best PSO for subpopulation evolution

After clustering, the population *pop* is clustered into *Num* subpopulations $subpopSet = \{subpop_k\}\ k = 1, \cdots, Num$, hereby each subpopulation has $Ns = N/Num$ particles $subpop_k = \{subpop_k^i\}$, $i = 1, \cdots, Ns$, where $subpop_k^i$ represents the *i*th particle of the *k*th subpopulation. Then, a personal archive P_k^i is formed to archive the historical information of

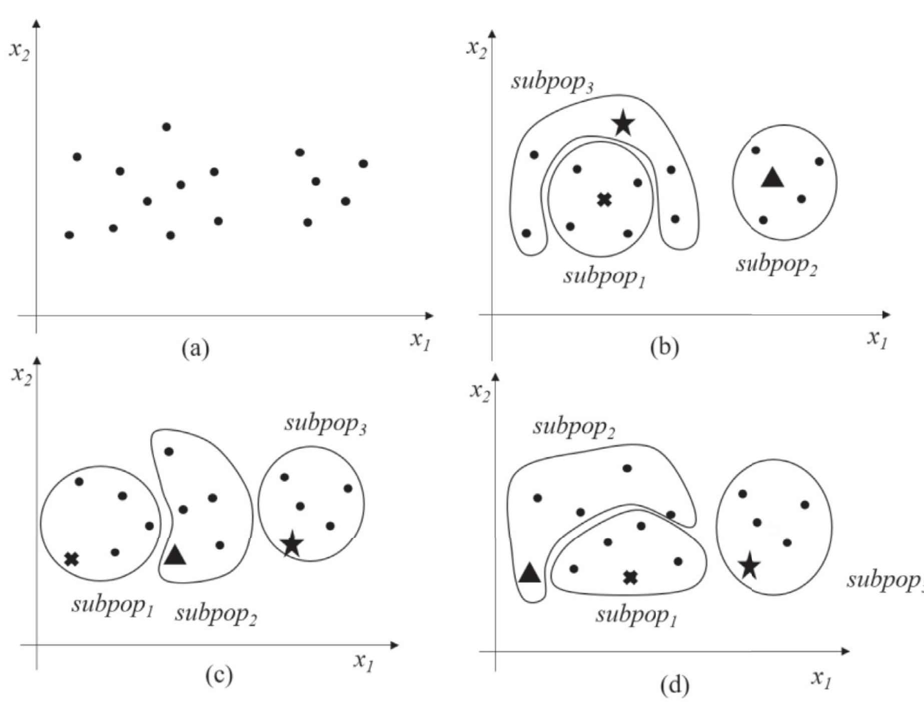


Fig. 3. Illustration of the effect of the chosen dimension in decision variable clustering method.

particle $subpop_k^i$. Next, the non-dominated solutions in subpopulation $subpop_k$ are saved into $NondomiSet_k$ and the particle with the largest special crowding distance in $NondomiSet_k$ is marked as $subGbest_k$. Particle $subpop_k^i$ updates its velocity and position guided by the personal best $pbest_k^i$ and global best $subGbest_k$ according to Equations (2) and (3). After evaluation, the updated particle $subpop'^i_k$ is stored into P_k^i and all particles dominated by $subpop'^i_k$ are removed. Then, the personal best $pbest_k^i$ and global best $subGbest_k$ will be updated accordingly. However, the way to update personal best $pbest_k^i$ and global best $subGbest_k$ is different from the convention. More precisely, a leader updating mechanism is applied, in which the leaders (personal/global best) will be updated only when a better position is found. Following this rule, the personal best $pbest_k^i$ will remain unchanged until it is dominated by the particles in P_k^i. Otherwise, $pbest_k^i$ will be replaced by the first particle in P_k^i according to non-dominated sorting and special crowding distance. Using the same principle, global best $subGbest_k$ will remain unchanged until it is dominated by $pbest_k^i$. Otherwise, $subGbest_k$ will be replaced by $pbest_k^i$. Afterward, P_k^i is incorporated into $NondomiSet_k$ and the dominated particles are removed. The pseudo-code of PSO based subpopulation search algorithm is provided in Algorithm 3.

In our algorithm, the personal archive for each particle is maintained. Once the particle is updated to a new one, it is added into its archive and all particles dominated by the new particle are removed. The current personal best of a particle is replaced only if it is dominated by the particles in its archive. That is to say, the personal best is updated only if a better position is found. Moreover, the new personal best is the non-dominated solution in the historical personal archive set and with the highest special crowding distance rank. Through combining the information from the search trajectory of the particle, the chosen personal best position could guide the particle toward the local optimum effectively and efficiently. Same as the updating of personal best of each particle, the global best of each subpopulation is updated only if there is a personal best dominates the current global best one.

Algorithm 3
Global-best PSO for subpopulation evolution

for $k = 1 : Num$
  for $i = 1 : Ns$
    Update the velocity of particle $subpop_k^i$ by Eq. (2);
    Update the position of particle $subpop_k^i$ by Eq. (3);
    Evaluate the objective values of updated particle $subpop'^i_k$;
    $P_k^i = [P_k^i, subpop'^i_k]$;
    Sort P_k^i according to the non-dominated sorting scheme and special crowding distance;
    if $pbest_k^i$ is dominated by particles in P_k^i
      $pbest_k^i \leftarrow$ the first particle in sorted P_k^i;
    endif
    if $subGbest_k$ is dominated by $pbest_k^i$
      $subGbest_k = pbest_k^i$;
    endif
      $NondomiSet_k = NondomiSet_k \cup P_k^i$;
      $NondomiSet_k \leftarrow$ non-dominated particles in $NondomiSet_k$;
  endfor
Endfor

3.4. Ring topology based PSO for local search

The inspiration for MMO-CLRPSO is originated from r3pso [30]. In order to promote information interaction among subpopulations, a ring topology based PSO is implemented on the global best $subGbest_k$ of each subpopulation to further explore the search space. As shown in Fig. 4, the non-dominated particles $NondomiSet_k$, $(k = 1, \cdots, Num)$ of each subpopulations are linked end to end to construct a ring topology. Hereby the non-dominated particles in the kth subpopulation $NondomiSet_k$ will have the neighbors $NondomiSet_{k-1}$ and $NondomiSet_{k+1}$. The neighborhood of $NondomiSet_1$ the first subpopulation contains $NondomiSet_{Num}$ of the last subpopulation, $NondomiSet_1$ and $NondomiSet_2$ of the second subpopulation, while the neighborhood of $NondomiSet_{Num}$ in the last subpopulation contains $NondomiSet_1$ of the first subpopulation, $NondomiSet_{Num}$ and $NondomiSet_{Num-1}$ of the $(Num-1)$th subpopulation. The non-dominated particles in the neighborhood of $NondomiSet_k$ are deposited into NBA_k. Next, particles are sorted according to non-dominated sorting mechanism and special crowding distance. Then $subGbest_k$ is updated through Equations (3) and (4), in which the first one in the sorted NBA_k is chosen as the historical best position of neighborhood and the first one in $NondomiSet_k$ is chosen as the historical personal best position. Afterward, the updated particles $subGbest'_k$ will be added into $NondomiSet_k$. Next, particles in $NondomiSet_k$ are sorted according to non-dominated sorting and special crowding distance. The dominated particles are removed and the one in the first rank is used to update $subGbest_k$. The iterations repeat until the entire global best $subGbest_k$ of each subpopulation are updated. The pseudo-code of ring topology based PSO for local search is present in Algorithm 4.

Algorithm 4
Ring topology based PSO for local search

for $k = 1 \cdots Num$
  if $k = 1$
    $Neighbor\{k\} = \{NondomiSet_1, NondomiSet_2, NondomiSet_{Num}\}$;
  elseif $k = Num$
    $Neighbor\{k\} = \{NondomiSet_{Num-1}, NondomiSet_{Num}, NondomiSet_1\}$;
  else
    $Neighbor\{k\} = \{NondomiSet_{k-1}, NondomiSet_k, NondomiSet_{k+1}\}$;
  endif
  Sort $Neighbor\{k\}$ according to the non-dominated sorting and special crowding distance;
  Set the first one in the sorted $Neighbor\{k\}$ as the historical best position of neighborhood;
  Set the first one in the sorted $NondomiSet_k$ as the historical personal best position;
  Update the velocity of particle $subGbest_k$ by Eq. (4);
  Update the position of particle $subGbest_k$ by Eq. (3);
  Evaluate the objective values of updated particle $subGbest'_k$;
  $NondomiSet_k = [NondomiSet_k, subGbest'_k]$;
  Sort $NondomiSet_k$ according to the non-dominated sorting and special crowding distance;
    $subGbest_k \leftarrow$ the first particle $NondomiSet_k$;
Endfor

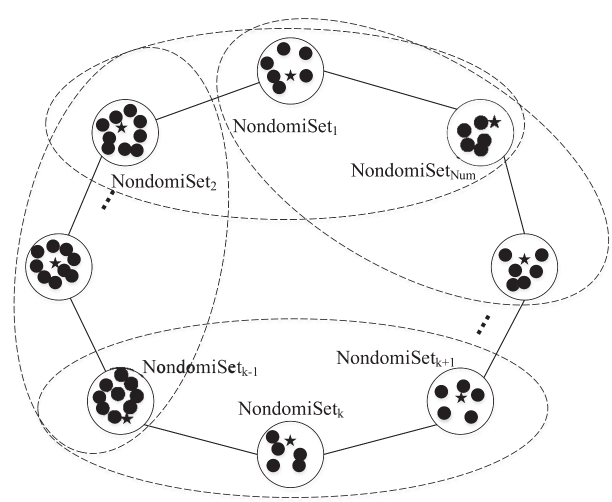


Fig. 4. Ring topology on non-dominated sets $NondomiSet_k$, $k = 1, \cdots, Num$. The black points indicate the non-dominated particles and the star indicates the best particle $subGbest_k$, $k = 1, \cdots, Num$.

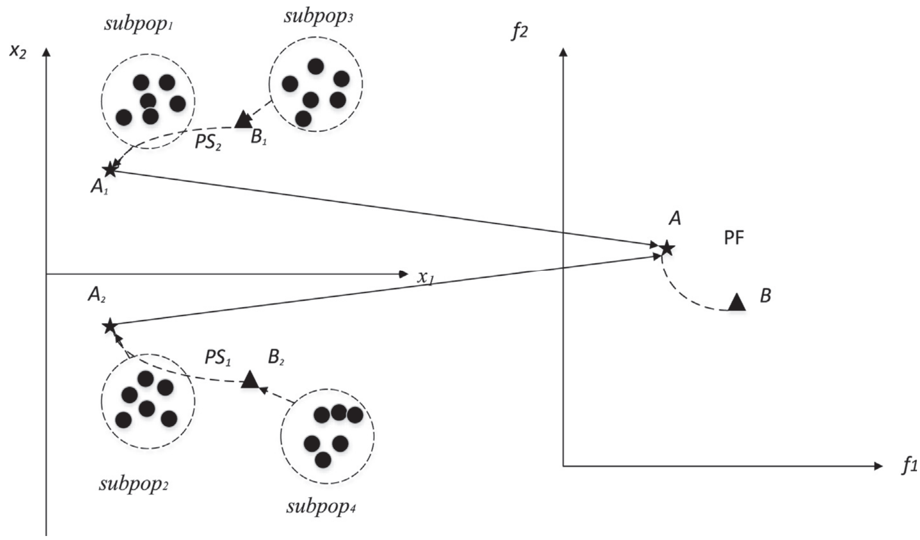


Fig. 5. Multimodal Multi-objective problem with multiple subpopulations. The point indicates a particle in subpopulation, the left stars is two true PSs in the decision space, while the right star is true PF.

When connecting the particles end to end by the ring topology, each particle could acquire guiding information from its neighbors which is benefit for maintaining diversity. However, the neighbor of particle is chosen almost randomly. Therefore, it may be detrimental regarding the convergence speed. In view of this, in our algorithms, multiple-population method is adopted for maintaining diversity and global-best PSO with leader updating mechanism of the subpopulation are used to guide the search in each subpopulation for a rapid convergence. However, subpopulations may be stagnating and hard to converge to the real optima without interacting with each other. Therefore, the ring topology is introduced to increase the interactions among subpopulations and avoid the stagnation. Instead of implementing on the whole population, the ring neighborhood topology is used to further improve the search on the non-dominated solutions of each subpopulation *GBA*.

3.5. Discussions

Multi-population mechanism is adopted in the proposed algorithm to handle the multimodality. It is expected that multiple solutions in the decision space could be located and preserved by separated subpopulations. As shown in Fig. 5, both A_1 and A_2 in the decision space correspond to the same individual A in the objective space, while both B_1 and B_2 in the decision space correspond to the same individual B in the objective space. Through clustering, multiple solutions in the decision space could be preserved at the same time with multiple separated subpopulations. As shown in Fig. 5, $subpop_1$ and $subpop_2$ will converge to solutions A_1 and A_2, respectively, while $subpop_3$ and $subpop_4$ will converge to the solutions B_1 and B_2 in the decision space, respectively.

Considering the multiple solutions in the decision space corresponding to the same Pareto optimal solution in the objective space, it has a great possibility that the solutions which are far away from each other will be more valuable than the closer ones. The reason is that the multiple PSs could provide various alternatives for the decision makers. If some of the acquired PSs become infeasible due to some reasons, the others could serve as a replacement. However, being affected by the same reason, those closer PSs will have higher probability to become infeasible than the PSs that are far away. Therefore, the PSs which are far away from each other in the decision space may be more valuable. Based on the decision variable clustering mechanism, the particles closer to each other will be clustered into the same subpopulation. In this way, the multiple PSs that are far away from each other would be preserved. However, the multiple PSs that are close to each other may get lost.

Similar to MO_Ring_PSO_SCD, the proposed MMO-CLRPSO is based on PSO and adopts the ring-topology as well. However, the way to implement these strategies is completely different. First, different from the single population adopted in MO_Ring_PSO_SCD, MMO-CLRPSO is based on multi-population mechanism, in which multiple Pareto optimal solutions in the decision space is designed to be tracked and preserved by different subpopulations independently at the same time. Second, PSO with the ring topology provides the main search power in MO_Ring_PSO_SCD, while ring topology is implemented on non-dominated solutions for the interaction among subpopulations to enhance the exploration and avoid the stagnation in our proposed MMO-CLRPSO. Third, when implementing PSO, instead of updating the leaders once the new particle is non-dominated, we propose a new leader updating mechanism, in which the leaders are updated only when the leader is dominated. The new leader updating mechanism is beneficial for avoiding the frequent replacement, keeping the mutual non-dominated solutions and accelerating convergence.

3.6. Complexity analysis

Considering the time and space complexity, the introduction of the strategies of clustering and leader updating mechanism of MMO-CLRPSO will consume memory space and increase the time complexity. The complexity of the decision variable clustering method is $O\left(\frac{N^2}{Num}\right)$, where *Num* is the number of the clusters, and the complexity of global-best PSO with leader updating mechanism is $O(MN)$, where M is the number of objective functions. The computational complexity of local-best PSO with ring-topology which is implemented on the non-dominated solutions of each subpopulation is $O\left(M\frac{N}{Num}\right)$. Overall, considering the *Num* is a constant and $M \prec\prec N$, the time complexity of the proposed algorithm is $O(N^2)$, which is smaller than that of NSGA-II and Omni-optimizer but slightly larger than that of MO_Ring_PSO_SCD. The space complexity of leader updating mechanism is $O(N)$. In summary, the consumption of the time and space is affordable.

4. Experimental study

In the section, experimental studies are conducted to evaluate the performance of the proposed MMO-CLRPSO. At first, a standard benchmark function and the performance indicator about MMO are introduced. After that, we discuss the influence of the different parameters and strategies to the performance of the proposed design. Finally, the proposed MMO-CLRPSO is compared with the other state-of-the-art algorithms.

4.1. Benchmark function and performance indicator

4.1.1. Benchmark function

Recently, Liang [6] summarized the previous works in this technical field and designed a set of multimodal multi-objective benchmark functions to test the performance of the algorithms. The benchmark function set [42] includes eleven test functions: MMF1, MMF2, MMF3, MMF4, MMF5, MMF6, MMF7, MMF8, SYM-PART simple, SYM-PART rotated, and Omni-test. These test functions inherit both multiple-objectives and multiple modality properties, i.e. they have one PF with more than one PS. Corresponding to the one PF, MMF1, MMF2, MMF3 and MMF7 have two PSs, MMF4, MMF5, MMF6 and MMF8 have four PSs, SYM-PART simple and SYM-PART rotated have nine PSs and Omni-test has twenty-seven PSs. Moreover, the PSs of some test functions overlap in every dimension (MMF3, MMF6, SYM-PART simple, SYM-PART rotated, Omni-test). Let us take MMF6 as an example to describe the test function in detail. For MMF6, the test function has two decision variables x_1 and x_2, and two objective variables f_1 and f_2. The MMF6 function is shown in Equation (6).

$$\begin{cases} f_1 = |x_1 - 2| \\ f_2 = \begin{cases} 1 - \sqrt{|x_1 - 2|} + 2(x_2 - \sin(6\pi|x_1 - 2| + \pi))^2 & -1 \le x_2 \le 1 \\ 1 - \sqrt{|x_1 - 2|} + 2(x_2 - 1 - \sin(6\pi|x_1 - 2| + \pi))^2 & 1 < x_2 \le 2 \end{cases} \end{cases} \tag{6}$$

where $1 \le x_1 \le 3$ and $-1 \le x_2 \le 2$.

Its true PS and true PF are shown in Equations (7) and (8).

$$x_2 = \begin{cases} \sin(6\pi|x_1 - 2| + \pi) & -1 \le x_2 \le 1 \\ \sin(6\pi|x_1 - 2| + \pi) + 1 & 1 < x_2 \le 2 \end{cases} \tag{7}$$

$$f_2 = 1 - \sqrt{f_1} \tag{8}$$

where $0 \le f_1 \le 1$. The true PS and PF are illustrated in Fig. 6.

It is observed that there are four true PSs according to the same objective values and these PSs overlap in every dimension. It poses a great challenge to the traditional multi-objective optimization algorithms to search for the multiple PSs.

4.1.2. Performance indicator

HV [50] and IGD [51] are considered the main criteria to evaluate the performance of multi-objective optimization algorithms. In this paper, HV indicator simultaneously measures diversity and convergence by computing the volume between the obtained PF and a reference point, z^*, and IGD reflects the average of the minimum distance of each point in the true PF to a point in the obtained PF. An algorithm producing a PF with a larger HV or a smaller IGD is considered a better design. Owing to the multimodal multi-objective optimization problem is multi-objective optimization problems, the performance indicator of multi-objective optimization problem can also evaluate the performance of multimodal multi-objective optimization. HV and IGD evaluate the performance of algorithms which focuses on the objective space while ignores the decision space, Thus, a new indicator denoted as *Pareto Set Proximity* (PSP) [42] is applied to evaluate the similarity between the obtained PSs and the true PSs.

$$PSP = \frac{CR}{IGDX} \tag{9}$$

where IGDX reflects the average distance between the obtained PS and the true PS [17]. CR is the cover rate between the obtained PS and the true PS [42]. The larger the PSP is, the better performance is for the underlying design.

4.2. Experimental setting

For each experiment, 20 independent runs are implemented. For each run, 100 iterations are executed, which result in $N \times 100$ function evaluations. The experimental results are presented in terms of PSP and HV under each of the experimental configurations. To allow a fair comparison, the related parameters of all compared algorithm are set as suggested in their respective references [6,41,42].

There are several parameters involved: N (population size), the number of the clusters *Num* w, $c1$ and $c2$ (the weight factor and the velocity control parameters of PSO). We set these parameters according to prior experience [22,42] and some initial experiments.

Population size N is crucial for the performance of the algorithm. If the population size is too small, there will be inadequate diversity in the population. The initial population with a smaller size cannot cover the decision space sufficiently which leads to a poor performance. On the contrary, if the population size is too large, it would consume too much computational recourse. Therefore, a proper setup of population size is required. According to our experience and the discussions in the related works [6,42], the population size N is assigned to be 800 in our experimental study, which is consistent with what used in the chosen state-of-the-art peer competitors. Limited by the space, the results

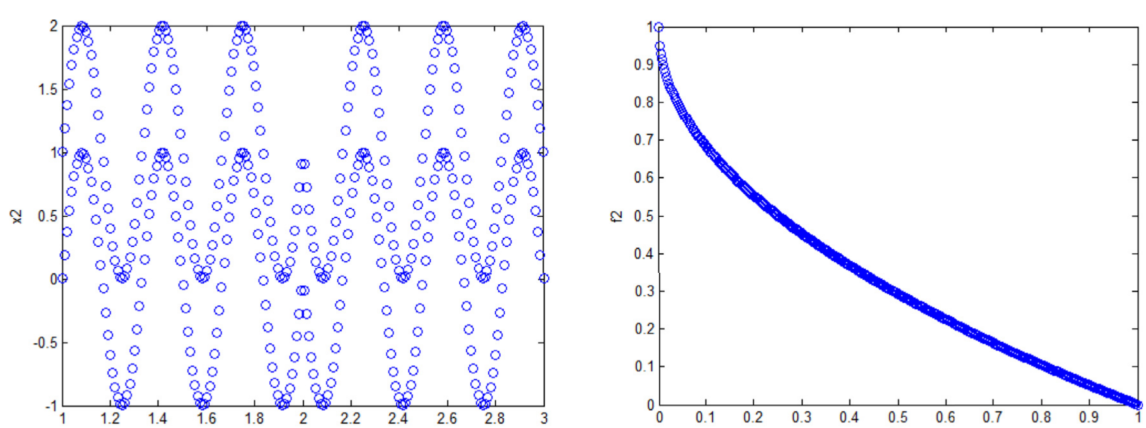


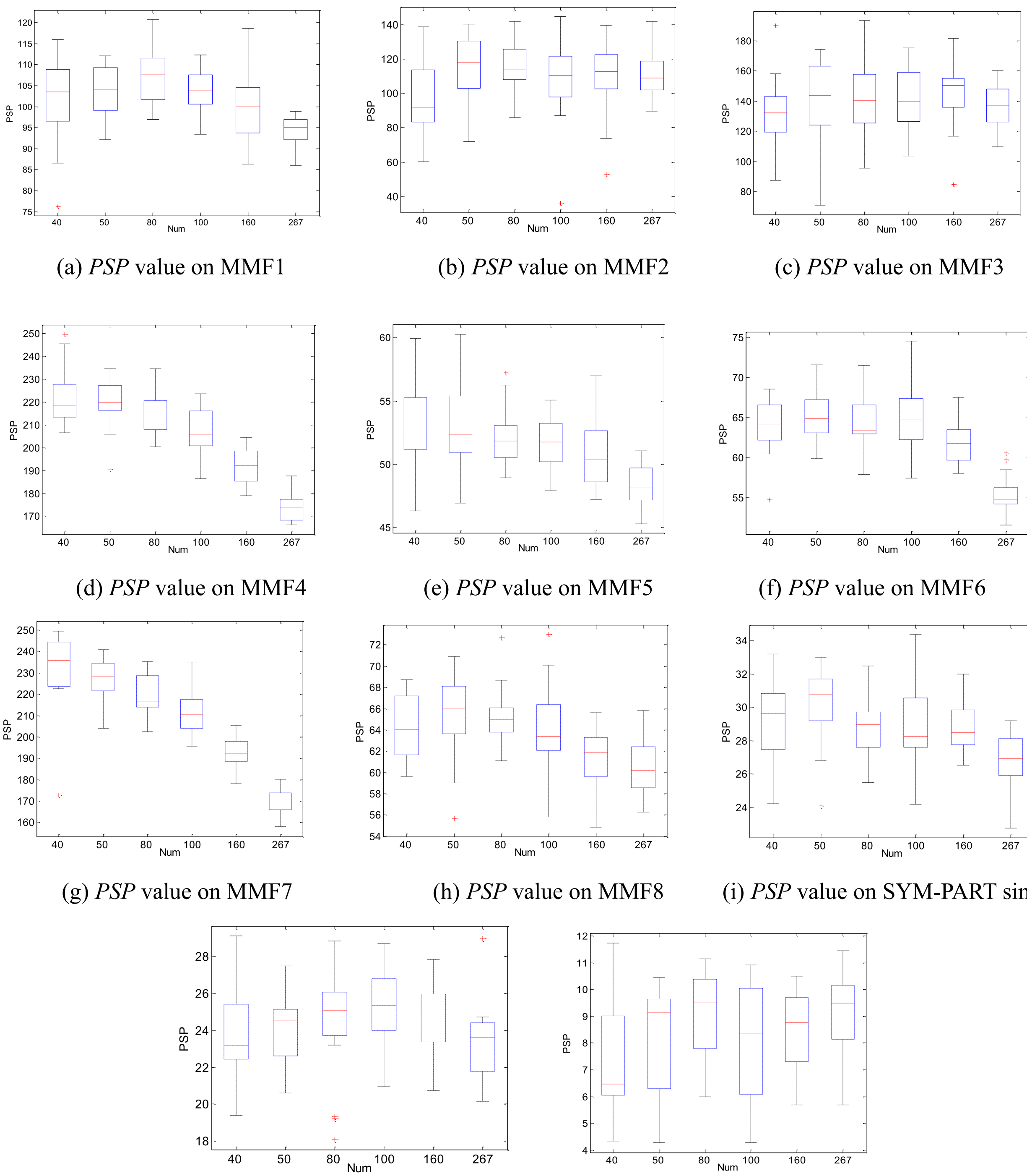
Fig. 6. The true PSs and the corresponding PF on MMF6.

regarding the varying population size is not presented. Instead, the number of the clusters *Num* is well analyzed in the following section. To the parameters in PSO, the frequently used setup in Ref. [30] is adopted in our work, in which the weight factor ***w*** is set to 0.7298, and both *c*1 and *c*2 are set to 2.05.

4.3. Analysis of involving mechanisms and related parameters

4.3.1. Analysis of the decision variable clustering method

A cluster method in the decision space is proposed to partition the population into multiple subpopulations. The number of clusters, *Num*, is a key parameter to the performance. With a fixed size population, the number of clusters *Num* could directly determine the size of each subpopulation in the decision space. In view of the evolution process of the



(a) *PSP* value on MMF1 (b) *PSP* value on MMF2 (c) *PSP* value on MMF3

(d) *PSP* value on MMF4 (e) *PSP* value on MMF5 (f) *PSP* value on MMF6

(g) *PSP* value on MMF7 (h) *PSP* value on MMF8 (i) *PSP* value on SYM-PART simple

(j) *PSP* value on SYM-PART rotated (k) *PSP* value on Omni-test

Fig. 7. The box-plots of PSP values of MMO-CLRPSO with different subpopulation sizes on eleven test problem. The horizontal axis of each plot set the number of subpopulation as 40, 50, 80, 100, 160, and 267, respectively.

subpopulation based on PSO, the size of the subpopulation is at least three. Each subpopulation aims to cover part of the decision space, too many particles in the subpopulation would be unnecessary. Therefore, the subpopulation size in the range of 3–20 is considered. According to the default setup of the whole population which is 800, the number of clusters is set to 40, 50, 80, 100, 160, and 267 in the simulation, and hereby the corresponding size of each cluster is 20, 16, 10, 8, 5 and 3, respectively. Note that the size of the last subpopulation is allowed to be less than *Ns* when the size of population cannot be divisible by the number of clusters.

Fig. 7 shows the PSP value with varying numbers of clusters on the 11 benchmark functions, where the horizontal axis represents the number of the clusters while the vertical axis represents the value of PSP for box-plots. It can be observed that the PSP value decreases on MMF4 and MMF7, first increase and then decrease on MMF1 and not fluctuate too much to the rest of the benchmarks with the increase of the number of clusters. For MMF1, MMF2, MMF6, MMF8, SYM-PART simple, SYM-PART rotated, Omni-test, the best performance of MMO-CLRPSO is when the number of subpopulation is 80. For MMF5, even though a better mean value could be observed when the number of subpopulations is either 50 and 40, the number of subpopulation set to 80 achieves more stable results. Overall, 80 is a good choice for the number of subpopulation. Moreover, owing to the values of HV with different numbers of clusters on the benchmark functions do not show much difference, the comparison of HV value is omitted here.

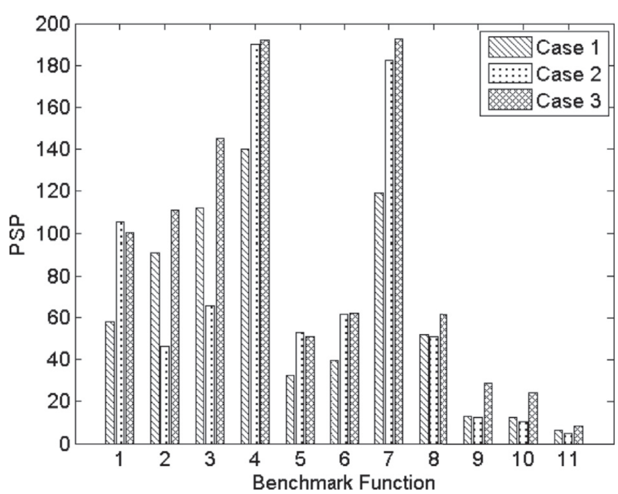


Fig. 8. The mean PSP of MMO-CLRPSO with proposed leader updating mechanism and MMO-CLRPSO with three leader updating strategies: case 1) replace the leader by the mutually non-dominating particle; case 2) according to 50% probability; case 3) do not update the leaders until they are dominated. The value on the horizontal axis of each plot indicate the following test function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = SYM-PART simple, 10 = SYM-PART rotated 11 = Omni-test.

4.3.2. Discussions of leader updating strategy of PSO

In multi-objective optimization, the way to update leaders is not as straightforward as in the single-objective optimization. There are three choices when the newly generated particles neither dominate nor be dominated by the current leaders: 1) replace the leader by the mutually non-dominating particle [27]; 2) according to some rule, e.g., a certain probability [28]; 3) do not update the leaders until they are dominated. Fig. 8 shows the comparison of the proposed algorithm with these three leader updating strategies, 50% probability is adopted for case 2.

It is observed that the performance of the algorithm with leader updating strategy case three achieves the best performance on 9 out of 11 functions, and is slightly worse than case two on the remaining two functions, MMF1 and MMF5. Replacing the leader by the mutually non-dominating particle seems to be the worst choice. Fig. 9 presents the comparison of the PS of the proposed algorithm with updating strategy of case one and three on function MMF5. It is observed that the non-dominated solutions in the decision space distribute more evenly with leader updating strategy case three than case one. The reason is there is frequent replacement of the leaders during the evolving process in the traditional PSO, i.e., the direction to guide particles changes frequently which will lead to the meaningless wandering and the slow convergence speed. In case three, the leaders only change if a better position among all the historical found position is found. In doing so, the particle will have a high speed to converge to the non-dominated front.

In addition, we also test the proposed leader updating strategy with pure PSO, i.e., there is no clustering and the local search method, and only the global-best PSO is applied. Fig. 10 presents the comparing results. It is observed that updating the leaders until better positions are found could obtain much better results than replacing the leaders with mutually non-dominating particles on all of the test functions.

4.3.3. Discussion of ring topology based local search

We compare the algorithm with and without the ring topology based local search in Fig. 11. It is observed that the algorithm with the ring topology based local search could obtain better results. The information

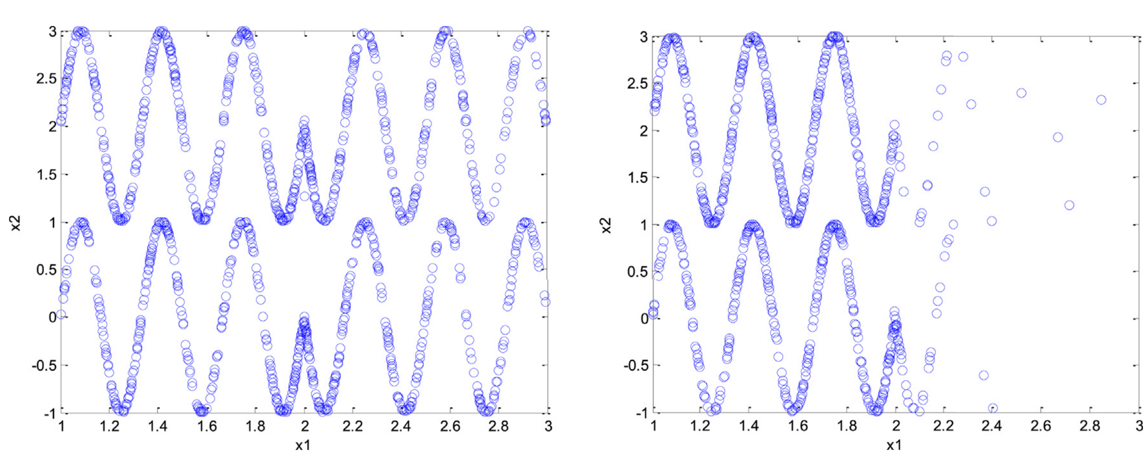


Fig. 9. The PS of the proposed algorithm with updating strategy case 3 in the decision space on the left for MM5, and the PS of MMO-CLRPSO with updating strategy case 1 in the decision space on the right for MM5.

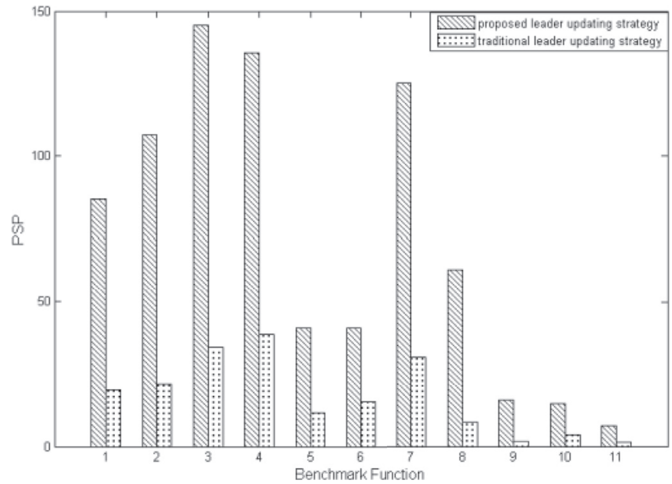


Fig. 10. The mean PSP of PSO with proposed leader updating mechanism, and PSO with the traditional leader updating strategy in a population. The value on the horizontal axis of each plot indicate the following test function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = SYM-PART simple, 10 = SYM-PART rotated 11 = Omni-test.

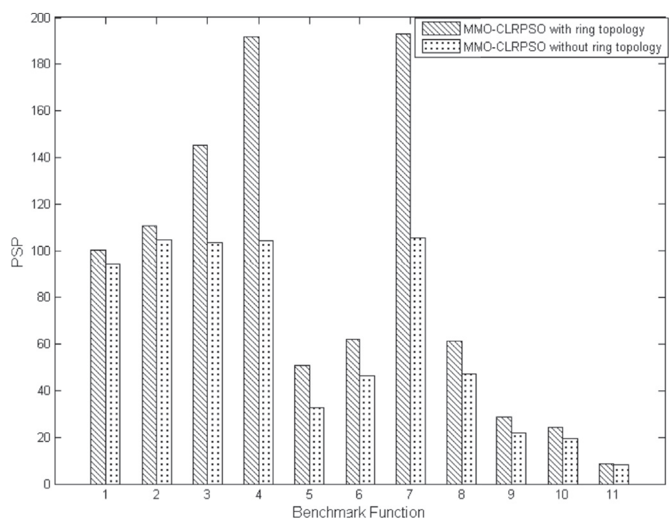


Fig. 11. The mean PSP of MMO-CLRPSO with and without a ring topology based on local search on eleven functions. The values on the horizontal axis of each plot indicate the following test function: 1 = MMF1, 2 = MMF2, 3 = MMF3, 4 = MMF4, 5 = MMF5, 6 = MMF6, 7 = MMF7, 8 = MMF8, 9 = SYM-PART simple, 10 = SYM-PART rotated 11 = Omni-test.

interaction among subpopulations is introduced by the ring-topology based PSO. Through the interaction among the subpopulations, the exploration ability could be significantly enhanced.

4.4. Comparison with other algorithms

In this section, we compare the proposed MMO-CLRPSO with six state-of-the-art algorithms including Omni_optimizer [41], DN-NSGAII [6], MO_Ring_PSO_SCD [42], SMPSO-MM [44], MMOPIO [45] and MMODE [47] on the 11 MMO benchmark functions considered. For a fair comparison, the compared algorithms are running 20 times on an *HP* computer (i7-8700, 8 GB), and all compared algorithms adopt the same parameter values to achieve the best performance. The size of population is set as 800 and the maximal number of function evaluations is set to 80,000. The PSP and HV values are shown in Fig. 12 and Table 1, respectively.

In Fig. 12, it is observed that the proposed MMO-CLRPSO achieves the highest PSP mean values on 6 out of 11 test functions (i.e., MMF1, MMF4, MMF5, MMF6, MMF7 and MMF8) among all the competing algorithms. Especially, the worst results of MMO-CLRPSO are better than the best results of the others on MMF4, MMF5, MMF6 and MMF7. MMO-CLRPSO defeats Omni-optimizer, DN-NSGAII and MO_Ring_PSO_SCD on almost all the test functions considered. However, MMO-CLRPSO is surpassed by MMOPIO, MMODE and MMO-CLRPSO on MMF2, MMF3 and SYM-PART simple. For SYM-PART rotated, MMO-CLRPSO obtains the better PSP mean values than that of SMPSO-MM, but worse than those of MMOPIO and MMODE. For Omni-test, MMO-CLRPSO is superior to the other algorithms except MMOPIO. It is observed that even higher mean values are obtained on some of the test functions, both SMPSO-MM and MMOPIO fluctuates considerably, especially on MMF2. MMF3, MMF8 and Omni-test. In conclusion, comparing with six state-of-the-art algorithms, MMO-CLRPSO obtains the best performance for PSP indicator on MMF1, MMF4, MMF5, MMF6, MMF7 and MMF8, and it could maintain a quite stable performance on the test functions even through it underperforms with respect to MMOPIO and MMODE on MMF2, MMF3, SYM-PART simple and SYM-PART rotated.

In Table 1, it is observed that the HV values of all the compared algorithms are very close to each other. MMO-CLRPSO achieves the best values on MMF8; Omni-optimizer obtains the highest HV values on MMF1, MMF2, MMF3, MMF5, MMF6, SYM-PART simple and Omni-test; the HV values of DN-NSGAII on SYM-PART simple and SYM-PART rotated are the highest among all algorithms. Comparing with DN-NSGAII, MMO-CLRPSO is surpassed on MMF2, MMF3, MMF5, SYM-PART simple, SYM-PART rotated, and Omni-test. Comparing with MO_Ring_PSO_SCD, MMO-CLRPSO obtained higher HV values on MMF4, SYM-PART simple, SYM-PART rotated, and Omni-test. Comparing with Omni-optimizer and DN-NSGAII, MMO-CLRPSO is not as superior, but its performance is comparable to each other according to HV values. Comparing with SMPSO-MM, MMOPIO and MMODE, the HV values of MMO-CLRPSO are very similar to them on all the test functions. As multimodal multi-objective algorithms, MMO-CLRPSO, MO_Ring_PSO_SCD, SMPSO-MM, MMOPIO and MMODE need to maintain the distribution of the population in the decision space. Therefore, little performance drop on the HV value would be acceptable.

5. Conclusions and future work

In this paper, MMO-CLRPSO is proposed to solve multimodal multi-objective optimization problems. Decision variable clustering method is proposed to divide the population into multiple subpopulations. Global-best PSO is adopted to conduct the exploitation in each of the subpopulation independently. After the non-dominated solutions of each subpopulation are obtained, the ring-topology based local search is implemented to further improve the exploration through the interaction among the found non-dominated solutions of each subpopulation. In this way, the independent evolution of each subpopulation and the interaction among the found non-dominated solutions of each subpopulation provide a good balance between the exploration and exploitation. Moreover, a new leader updating strategy is proposed to replace the traditional leader updating mechanism in the PSO. The experimental results have shown that the decision variable clustering method could help locate multiple PSs in the decision space, the new leader updating strategy could improve the convergence efficiency, and the ring-topology based search could enhance the interaction among subpopulations to improve the exploration effectiveness. The experimental results show the effectiveness of the proposed algorithm.

Considering the real-world applications, it would be practical to provide one or few robust solutions both in the decision space and the

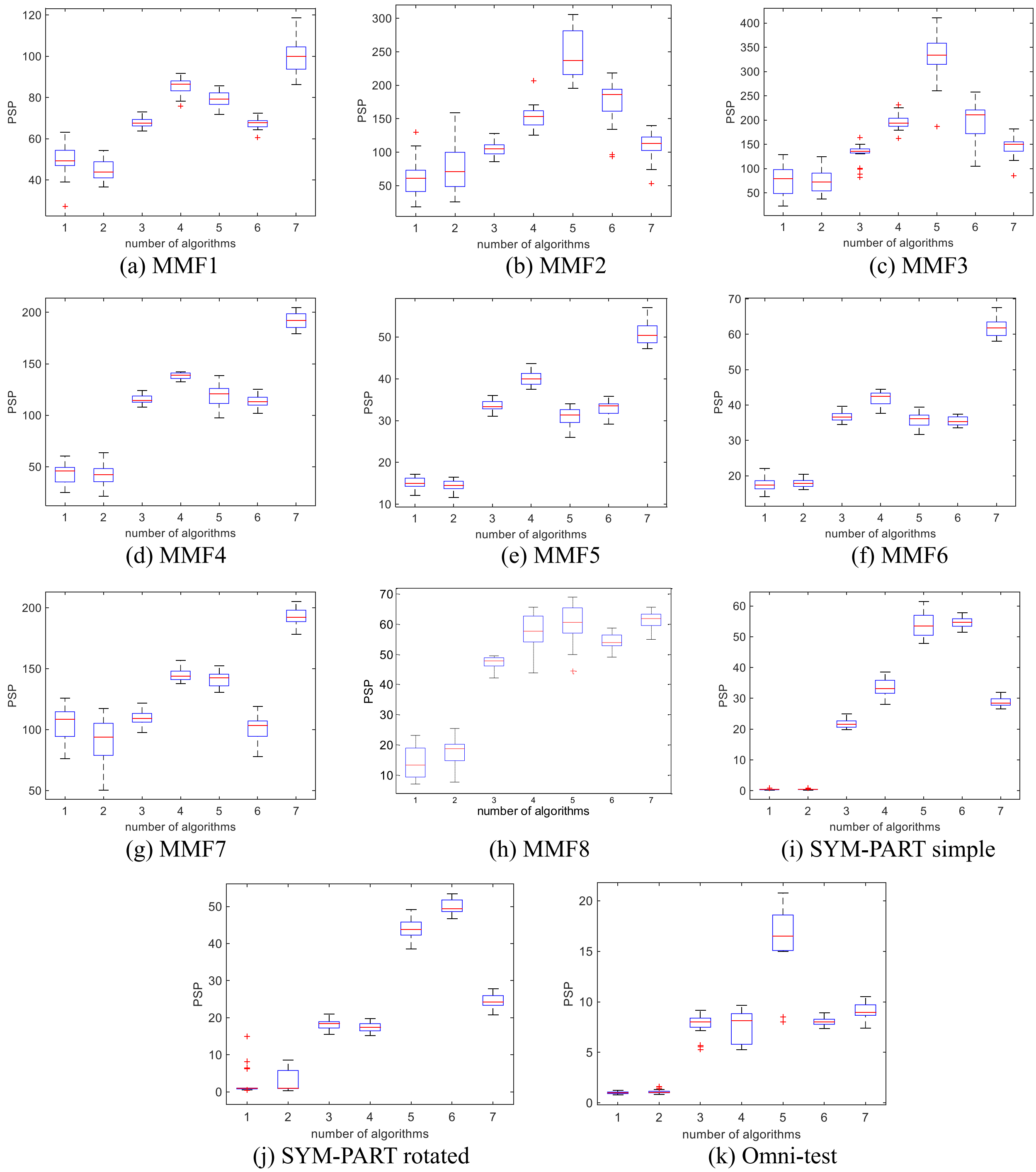


Fig. 12. The box-plots of PSP value of different algorithms on eleven test functions. The numerals on the horizontal axis of each plot indicate the following algorithms: 1 = Omni_optimizer [41], 2 = DN-NSGAII [6], 3 = MO_Ring_PSO_SCD [42], 4 = SMPSO-MM [44], 5 = MMOPIO [45], 6 = MMODE [47], and 7 = MMO-CLRPSO.

Table 1
HV value of different algorithms.

Function	Omni-optimizer [41]	DN-NSGAII [6]	MO_Ring_PSO_SCD [42]	SMPSO-MM [44]	MMOPIO [45]	MMODE [47]	MMO-CLRPSO
MMF1	**3.67 ± 2.43e-05**	3.66 ± 1.12e-03	3.66 ± 4.54e-04	3.67 ± 5.43e-04	3.66 ± 6.12e-04	3.66 ± 3.01 e−04	3.66 ± 6.00e-04
MMF2	**3.67 ± 4.69e-05**	3.66 ± 9.08e-03	3.65 ± 7.61e-03	3.66 ± 3.80e-03	3.66 ± 3.4e-03	3.65 ± 6.6 e−03	3.65 ± 4.3e-03
MMF3	**3.67 ± 2.52e-05**	3.67 ± 5.25e-04	3.65 ± 6.63e-03	3.66 ± 2.2e-03	3.66 ± 3.0e-03	3.65 ± 4.5 e−03	3.65 ± 6.3e-03
MMF4	3.32 ± 3.84e-05	3.18 ± 3.31e-04	3.30 ± 9.54e-04	3.33 ± 7.92e-04	**3.33 ± 5.98e-04**	3.32 ± 1.8 e−03	3.33 ± 8.85e-03
MMF5	**3.67 ± 1.81e-05**	3.67 ± 3.22e-04	3.66 ± 3.89e-04	3.66 ± 3.65e-04	3.66 ± 6.29e-04	3.66 ± 3.12e-04	3.66 ± 3.78e-04
MMF6	**3.67 ± 2.19e-05**	3.66 ± 1.41e-03	3.66 ± 3.33e-04	3.66 ± 2.85e-04	3.66 ± 4.96e-04	3.66 ± 4.56e-04	3.66 ± 3.54e-04
MMF7	**3.67 ± 4.44e-05**	3.66 ± 1.24e-03	3.67 ± 2.10e-04	3.66 ± 2.52e-04	3.66 ± 2.29e-04	3.66 ± 5.73e-04	3.66 ± 2.65e-04
MMF8	3.21 ± 1.20e-04	3.21 ± 1.22e-03	3.21 ± 1.09e-03	3.21 ± 1.1e-03	3.21 ± 2.89e-04	3.21 ± 9.14e-04	**3.21 ± 1.16e-04**
SYM-PART simple	1.32 ± 3.21e-04	**1.32 ± 1.94e-04**	1.30 ± 1.65e-03	1.31 ± 1.71e-04	1.32 ± 2.46e-04	1.32 ± 2.85 e−04	1.32 ± 6.33e-04
SYM-PART rotated	1.32 ± 2.55e-04	1.32 ± 4.49e-04	1.29 ± 3.55e-03	1.29 ± 2.4e-03	1.30 ± 4.85e-04	**1.32 ± 2.49 e-04**	1.31 ± 1.2e-03
Omni-test	**62.06 ± 2.46e-04**	62.06 ± 3.95e-04	61.93 ± 2.14e-01	61.97 ± 1.01e-02	62.05 ± 3.20e-04	48.13 ± 2.07e-04	61.99 ± 8.2e-03

objective space. Therefore, considering the robustness of the solutions when handling the multimodal multi-objective optimization problems will be our future work.

Acknowledgment

The work is supported by the National Natural Science Foundation of China (No. 61403349, 61501405), key scientific research projects of universities in Henan province (No. 18A210025, 20A520004), science and technology key project of Henan province (No. 182102110399, 192102110203) and Postgraduate Technology Innovation Project in Zhengzhou University of Light Industry (No. 2017032).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.swevo.2019.100569.

References

[1] M.G. Gong, H. Li, E.H. Luo, J. Liu, J. Liu, A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing, IEEE Trans. Evol. Comput. 21 (2) (2017) 234–248.
[2] M.G. Gong, Z. Wang, Z.X. Zhu, L.C. Jiao, A similarity-based multiobjective evolutionary algorithm for deployment optimization of near space communication system, IEEE Trans. Evol. Comput. 21 (6) (2017) 878–897.
[3] F. Altiparmak, M. Gen, L. Lin, T. Paksoy, A genetic algorithm approach for multi-objective optimization of supply chain networks, Comput. Ind. Eng. 51 (1) (2006) 196–215.
[4] D. Rodolfo, B. José, M. José, A. José, J. Ignacio, L. Juan, A. Ismael, Multi-objective optimization minimizing cost and life cycle emissions of stand-alone PV-wind-diesel systems with batteries storage, Appl. Energy 88 (11) (2011) 4033–4041.
[5] E. Asadi, M.G. Da Silva, C.H. Antunes, L. Dias, Multi-objective optimization for building retrofit strategies: a model and an application, Energy Build. 44 (2012) 81–87.
[6] J. Liang, C.T. Yue, B.Y. Qu, in: 2016 IEEE Congress on Evolutionary Computation, CEC 2016, Institute of Electrical and Electronics Engineers Inc, 2016, pp. 2454–2461.
[7] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248.
[8] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[9] B. Qu, P.N. Suganthan, J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.
[10] Z. Ender, M. Lmaz, Particle swarms for multimodal optimization, in: International Conference on Adaptive and Natural Computing Algorithms, 2007, pp. 366–375.
[11] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application, 1987, pp. 41–49.
[12] T. Li, C. Wei, W. Pei, PSO with sharing for multimodal function optimization, in: International Conference on Neural Networks and Signal Processing, 2003, pp. 450–453.
[13] K. Dejong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph. D. Thesis, University of Michigan, 1975.
[14] M.Q. Li, D. Lin, J.S. Kou, A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization, Appl. Soft Comput. J. 12 (3) (2012) 975–987.
[15] A. Petrowski, A clearing procedure as a niching method for genetic algorithms, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 798–803.
[16] J. Li, M.E. Balazs, G.T. Parks, P. Clarkson, Erratum, A species conserving genetic algorithm for multimodal function optimization, Evol. Comput. 10 (3) (2002) 207–234.
[17] A. Zhou, Q. Zhang, Y. Jin, Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm, IEEE Trans. Evol. Comput. 13 (5) (2009) 1167–1189.
[18] H. Xia, J. Zhuang, D.H. Yu, Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization, IEEE Trans. Cyber. 44 (3) (2014) 378–393.
[19] C.R. Raquel, P.C. Naval, An effective use of crowding distance in multi-objective particle swarm optimization, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, 2005, pp. 257–264.
[20] Z. Li, J. Tan, Y.X. Feng, Multi-objective particle swarm optimization algorithm based on crowding distance sorting and its application, Comput. Integr. Manuf. Syst. (2008) 1329–1336.
[21] B. István, A cluster-based evolutionary algorithm for multi-objective optimization, computational intelligence, Theory Appl. (2001) 357–368.
[22] B.Y. Qu, J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inf. Sci. 197 (2012) 131–143.
[23] G.H. Zhang, X.Y. Shao, P.G. Li, L. Gao, An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, Comput. Ind. Eng. 56 (4) (2009) 1309–1318.
[24] D. Liu, K.C. Tan, C.K. Goh, W.K. Ho, A multi-objective memetic algorithm based on particle swarm optimization, IEEE Trans. Syst. Man Cybern. B Cybern. 37 (1) (2007) 42–50.
[25] L. Li, W.L. Wang, X.L. Xu, Multi-objective particle swarm optimization based on global margin ranking, Inf. Sci. 375 (2017) 30–47.
[26] Q.L. Zhu, Q.Z. Lin, W.N. Chen, et al., An external archive-guided multiobjective particle swarm optimization algorithm, IEEE Trans. Cyber. 47 (9) (2017) 2794–2808.
[27] C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.
[28] J.J. Liang, B.Y. Qu, P.N. Suganthan, B. Niu, Dynamic Multi-Swarm Particle Swarm Optimization for Multi-Objective Optimization Problems, 2012, IEEE Congress on Evolutionary Computation, Brisbane, QLD, 2012, pp. 1–8.
[29] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evol. Comput. 39 (2018) 24–35.
[30] X.D. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14 (1) (2009) 150–169.
[31] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, an overview, Swarm Intell. 1 (2007) 33–57.
[32] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (5) (2002) 443–462.
[33] S. Alam, G. Dobbie, Y.S. Koh, P. Riddle, S.U. Rehman, Research on particle swarm optimization based clustering: a systematic review of literature and techniques, Swarm Evol. Comput. 17 (2014) 1–13.
[34] M.R. Bonyadi, Z. Michalewicz, A locally convergent rotationally invariant particle swarm optimization algorithm, Swarm Intell. 8 (3) (2014) 159–198.
[35] H.Z. Yang, F.C. Li, C.M. Wang, A density clustering based niching genetic algorithm for multimodal optimization, in: IEEE International Conference on Machine Learning and Cybernetics, 2005, pp. 1599–1604.
[36] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (1988) 95–99.
[37] F. di Pierro, S.T. Khu, D.A. Savić, An investigation on preference order ranking scheme for multi-objective evolutionary optimization, IEEE Trans. Evol. Comput. 11 (1) (2007) 17–45.
[38] S.X. Yang, M. Li, X.H. Liu, J.H. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736.
[39] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (1) (2016) 16–37.
[40] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[41] K. Deb, S. Tiwari, Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization, Eur. J. Oper. Res. 185 (3) (2008) 1062–1087.
[42] C.T. Yue, B.Y. Qu, J. Liang, A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems, IEEE Trans. Evol. Comput. 22 (5) (2008) 805–817.
[43] O. Schütze, M. Vasile, C.a. Coello Coello, Computing the set of epsilon-efficient solutions in multiobjective space mission design, J. Aerosp. Comput. Inf. Commun. 8 (2011) 53–70.
[44] J. Liang, Q. Guo, C. Yue, A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems, in: Proceedings of International Conference on Swarm Intelligence, 2018, pp. 550–560.
[45] Y. Hu, J. Wang, J. Liang, K. Yu, H. Song, Q. Guo, C. Yue, A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm, Sci. China Inf. Sci. 62 (70206) (2019) 1–17, https://doi.org/10.1007/s11432-018-9754-6. Submitted for publication.
[46] Y. Liu, G.G. Yen, D. Gong, A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies, IEEE Trans. Evol. Comput. (2018). Article in press.
[47] J. Liang, W. Xu, C. Yue, K. Yu, H. Song, Oscar D. Crisalle, Multimodal multiobjective optimization with differential evolution, Swarm Evol. Comput. 44 (2019) 1028–1059.
[48] S.X. Yang, C.H. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput. 14 (6) (2010) 959–974.
[49] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, A competitive clustering particle swarm optimizer for dynamic optimization problems, Swarm Intell. 6 (3) (2012) 177–206.
[50] Z. Eckart, T. Lothar, Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.
[51] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2) (2003) 174–188.