A Multi-objective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multi-objective Problems

Caitong Yue, *Student Member*, *IEEE*, Boyang Qu, *Member*, *IEEE* and Jing Liang, *Member*, *IEEE*

***Abstract*—This paper presents a new particle swarm optimizer for solving multimodal multi-objective optimization problems which may have more than one Pareto-optimal solution corresponding to the same objective function value. The proposed method features an index-based ring topology to induce stable niches that allow the identification of a larger number of Pareto-optimal solutions, and adopts a special crowding distance concept as a density metric in the decision and objective spaces. The algorithm is shown to not only locate and maintain a larger number of Pareto-optimal solutions, but also to obtain good distributions in both the decision and objective spaces. In addition, new multimodal multi-objective optimization test functions and a novel performance indicator are designed for the purpose of assessing the performance of the proposed algorithms. An effectiveness validation study is carried out comparing the proposed method with five other algorithms using the benchmark functions to prove its effectiveness.**

***Index Terms*—Multimodal, multi-objective, particle swarm optimization, ring topology, benchmark functions.**

I. INTRODUCTION

MULTI-OBJECTIVE optimization problems have two or more conflicting objectives to be optimized. Without loss of generality, the minimization multi-objective problem can be formulated as:

$$\min \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})] \quad (1)$$

subject to:

$$g_i(\vec{x}) \le 0 \quad i = 1, 2, \dots, k \quad (2)$$

$$h_j(\vec{x}) = 0 \quad j = 1, 2, \dots, p \quad (3)$$

where $\vec{x} = (x_1, x_2, \dots, x_n)$ is an n-dimensional decision vector; $\vec{f}$ is an m-dimensional objective vector; $g_i(\vec{x}) \le 0\ (i = 1, 2, \dots, k)$ represent inequality constraints; and $h_i(\vec{x}) = 0\ (j = 1, 2, \dots, p)$ are equality constraints. The n-dimensional space consisting of all the possible values of the decision vector $\vec{x}$ is known as the *decision space*, and the m-dimensional space comprised of all the possible values of the objective vector $\vec{f}(x)$ is the *objective space*. In multi-objective optimization problems, different solutions can be compared according to the *Pareto dominance relationship*: Given two feasible solutions $\vec{x}$ and $\vec{y}$, solution $\vec{x}$ is said to dominate $\vec{y}$ if $f_i(\vec{x}) \le f_i(\vec{y})$ for $i = 1, \dots, m$ and there exists at least one $j \in \{1, \dots, m\}$ so that $f_j(\vec{x}) < f_j(\vec{y})$. A solution is said to be *non-dominated* if it is not dominated by any other solution. The set of all the non-dominated solutions in the decision space is called the *Pareto-optimal set* (PS). The *Pareto front* (PF) is the set of all the vectors in the objective space that correspond to the PS.

In multi-objective optimization problems there may exist two or more distinct PSs corresponding to the same PF. Arguably, finding one of these multiple PSs may be sufficient to obtain an acceptable solution for some problems. However, failing to identify more than one of the PSs may prevent the decision maker from considering solution options that could bring about improved performance. A simple real-world example is illustrated in the path-planning problem summarized in Fig. 1, where travelers desire to drive from a starting point to the destination in the least possible time and involving the fewest number of intersections. There are six non-dominated options, with objective values as described in the caption of Fig. 1. It is known that {Option 1, Option 2, Option 3} and {Option 4, Option 5, Option 6} are two PSs that correspond to the same PF. However, the communal facilities encountered by these paths are different, as the figure shows that the gas station is accessible along the paths in {Option 1, Option 2, Option 3}, whereas there are no gas stations in the paths {Option 4, Option 5, Option 6}. The gas station is necessary for some travelers, while some other travelers want to avoid the gas station since it is not safe sometimes. If an algorithm obtains only one PS, it cannot meet the needs of different travelers. Hence, in this scenario it is of significant value to maintain more than one PS for the purpose of further decision making.

A few real-world optimization problems with multiple PSs

The work is supported by National Natural Science Foundation of China (61473266, 61673404, 61305080, and U1304602), China Postdoctoral Science Foundation (No.2014M552013), Project supported by the Research Award Fund for Outstanding Young Teachers in Henan Provincial Institutions of Higher Education of China (2014GGJS-004) and Program for Science & Technology Innovation Talents in Universities of Henan Province in China (16HASTIT041). (*Corresponding author: Jing Liang*.)

Caitong Yue and Jing Liang are with Industrial Technology Research Institute and School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: zzuyuecaitong@163.com; liangjing@zzu.edu.cn).

Boyang Qu is with School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China (e-mail: qby1984@hotmail.com).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

have already been identified in the literature [1], [2]. An in-depth analysis of these prior works leads to the conclusion that in practical applications engineers can take advantage of knowledge of a larger number of Pareto-optimal solutions. Furthermore, the costs of alternative Pareto-optimal solutions may differ significantly [3]. Therefore, it is of significant practical benefit to develop effective algorithms to solve optimization problems that may have more than one PS that maps to the same PF. In 2016, Liang [4] referred to this class of problems as *multimodal multi-objective optimization problems.*

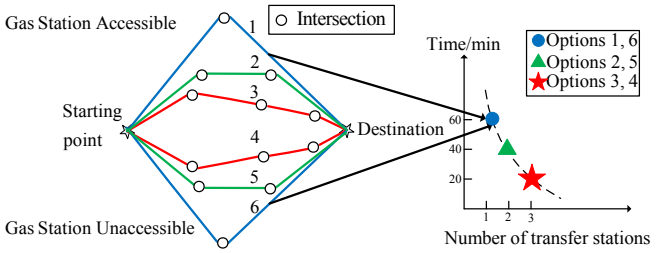


Fig. 1. Path planning problem. Option 1: 1 intersection, 60 min; Option 2: 2 intersections, 40 min; Option 3: 3 intersections, 20 min; Option 4: 3 intersections, 20 min; Option 5: 2 intersections, 40 min; Option 6: 1 intersection, 60 min.

Multimodal optimization refers to locating not only one optimum, but instead identifying a set of locally optimal solutions [5]. Traditionally, multimodal problems are posed as single-objective problems that may have more than one local optimum or several global optima. A most popular method to solve multimodal problems is niching technique [6] inspired on the way organisms evolve in nature. Numerous variations of niching methods have been proposed, including fitness sharing [7], [8], crowding [9], clearing [10] and speciation [11]. However, these niching methods involve a relatively large number of parameters, such as the sharing radius or crowding factor. These parameters are often difficult to set by the use of the algorithm, because they depend on specific characteristics of each optimization problem. To avoid introducing new parameters, Li [12] proposed an index-based ring-topology particle swarm optimization algorithm for multimodal optimization. Its main characteristic is that it can form stable niches without any niching parameters.

As a stochastic optimization algorithm, particle swarm optimization (PSO) is a robust and effective optimization technology [13] that was first proposed in 1995 [14] to solve single-objective optimization problems. Subsequently, many PSO variants have been proposed to solve multi-objective or multimodal problems [15]-[19]. This paper proposes a new PSO variant for multimodal multi-objective optimization problems.

When solving multimodal multi-objective optimization problems, particular attention should be paid to the decision space. Note that niches can be induced in the decision space to assist in searching for more Pareto-optimal solutions. In addition, in principle a crowding-distance metric can be embedded in the decision space to play a role in environmental selection processes. These issues are addressed in this paper through the introduction of a new algorithm that the authors call multi-objective particle swarm optimization using ring topology and special crowding distance (MO_Ring_PSO_SCD) conceived for solving multimodal multi-objective problems. Ring topology is adopted to form stable niches and locate multiple optima. All the solutions are sorted according to two indicators called *non-dominated ranking* and *special crowding distance.* The net effect is that a larger number of Pareto-optimal solutions can be maintained.

This work includes five major contributions. First, multimodal multi-objective problems are introduced, along with their dominant features, and a real-world application example is given as a practical illustration. Second, a novel multi-objective particle swarm optimization using ring topology and special crowding distance is proposed. The ring topology is effective for finding a larger number of Pareto-optimal solutions. Two elite archives are established to store the population history, and in addition, the special crowding distance acts as a second criterion for environmental selection, which helps to maintain more Pareto-optimal solutions and improve their diversity. Third, new multimodal multi-objective test functions are designed, along with technical treatments of the problem function, feasible region, true PF and PSs. Fourth, this paper proposes the concept of *Pareto sets proximity* (*PSP*) as a new performance indicator that reflects both the overlap ratio and distance between the true and the obtained PSs. Fifth, the effectiveness of the new proposed algorithm is verified by (*i*) comparing the performance of multi-objective particle swarm optimization algorithms with and without ring topology or special crowding distance, (*ii*) comparing the new algorithm with three state-of-the-art multi-objective algorithms and two multimodal multi-objective algorithms, and (*iii*) analyzing in the mechanics and convergence behavior of the MO_Ring_PSO_SCD in detail.

The reminder of this paper is organized as follows. Section II reviews related works. The proposed algorithm MO_Ring_PSO_SCD is described in Section III. Section IV proposes features and complexity of multimodal multi-objective test functions and Section V gives a novel performance indicator. Experiments and analysis are presented in Section VI. At last, the paper is concluded in Section VII.

II. Related Works

A. Prior Works on Manipulating the Distribution of the Solutions in the Decision Space

Prior researchers have developed tools for manipulating and analyzing the distribution of the solutions in the decision space. Deb [20] proposed the Omni-optimizer algorithm, where the concept of a crowding distance in the decision space was introduced and used to play a role in a non-dominated sorting scheme. It is straight forward to recognize that, maintaining good distributions is not equivalent to locating more Pareto-optimal solutions. To that end, a supplementary operation should be introduced. Chan [21] adopted the concepts of Lebesgue contribution and neighborhood count to maintain diversity in the decision and objective spaces. The radius of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

neighborhood, which plays a central role in this algorithm, is calculated as the maximum closest distance between the solution and its neighbors, and hence is a time consuming numerical operation.

Zhou [22] proposed a probabilistic model based on a multi-objective evolutionary algorithm to simultaneously approximate the PS and the PF. However, this algorithm can only deal with multi-objective problems of Class II [22]. Other drawback of the method is the technique performs poorly when the PS is a linear manifold.

Liang [4] defined the multimodal multi-objective optimization problem, which could be succinctly illustrated through the diagram in Fig. 2. The figure shows a simple diagram illustrating a case where there are two PSs corresponding to the same PF. Liang [4] proposed an algorithm called DN-NSGAII conceived for the purpose of locating more PSs. The technique first ranks solutions according to non-dominated sorting scheme and the solutions in the same front are then sorted based on a decision-space crowding distance. Therefore, in DN-NSGAII the non-dominated and less-crowded solutions in the decision space are preferred. It has been demonstrated that this algorithm can obtain more solutions than NSGAII [23] and various other commonly used multi-objective optimization approaches. However, the distribution of solutions in the decision space is not very good (*cf.* Fig. 9 (a) and (c) in [4]).

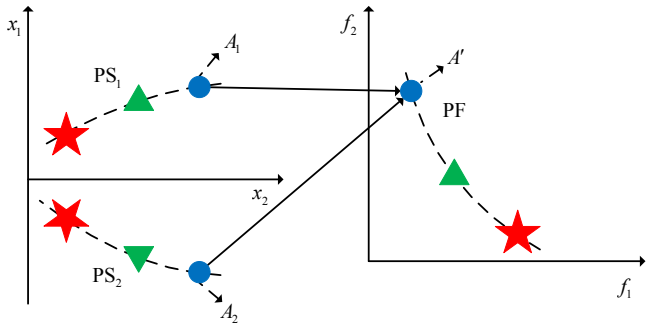


Fig. 2. Illustration of multimodal multi-objective problem.

B. The Framework of Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based algorithm, which makes it particularly well suited for solving multimodal optimization problems. PSO was inspired by the social behavior of birds within a flock [14]. In this method, the historical personal best position of a point (also called *particle*) is denoted *pbest*, and the historical best position of its neighborhood is called *nbest*. Each particle in the swarm is led by *pbest* and *nbest* to fly from a starting position to a better area. Let $\vec{x}_i(t)$ and $\vec{v}_i(t)$ denote the position and velocity of particle p_i of the t^{th} generation. They are updated according to the following equations

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (4)$$

and

$$\vec{v}_i(t) = W\vec{v}_i(t-1) + C_1 r_1(\vec{x}_{pbest_i} - \vec{x}_i(t)) + C_2 r_2(\vec{x}_{nbest_i} - \vec{x}_i(t)) \quad (5)$$

where W is the inertia weight (often set to 0.7298, according to [24]), C_1 and C_2 are constants (which satisfy the equality $C_1 + C_2 = 4.1$ [24]) used to balance exploration and exploitation processes, and r_1 and r_2 are random values uniformly generated in the range [0, 1]. The key step in the particle swarm optimization methodology is the selection of leaders for the current particles [25].

When extending PSO algorithms from the single-objective to the multi-objective case, three issues become of particular significance: (*i*) the selection of a leader particle, (*ii*) the management of the distribution properties, and (*iii*) the management of the convergence speed. Regarding the first issue, the leader-particle selection is a relatively straightforward procedure in single-objective optimization, where one can simply declare that particle with the largest fitness value is the best candidates to serve as leaders. In multi-objective optimization, however, the identification of a best leader among all the solutions may become a challenging task due to inherent conflicts introduced by the multiple objectives. In that case it is natural to give preference to non-dominated solutions to serve as candidates for the leader designation. Concerning the second issue, it is necessary to develop of a methodology that can ensure a good distribution of solutions in the decision space and in the corresponding objective space. Finally, regarding the third issue, the PSO convergence speed has been addressed through a number of communication topologies. In particular, the star, ring, and von Neumann topologies are known to be effective in avoiding premature convergence. The topology-based PSO algorithms are reviewed and compared in [26]. However, these topological approaches cannot induce stable niches, and as a consequence in multimodal problems the population converges to a single solution instead of multiple solutions. In contrast, r3pso described in [12], which incorporates an index-based ring topology, has been demonstrated through experimental results that it can form stable niches. In addition, the algorithm does not require the introduction of niching parameters, which is an attractive feature. The three issues described here are addressed in the new algorithm described in the next section.

III. Description of MO_Ring_PSO_SCD

As a population-based algorithm, particle swarm optimization has the capability, and hence the natural advantage of searching for multiple optima in a single run. In multimodal multi-objective optimization problems, where a large number of Pareto-optimal solutions should be searched for and maintained, PSO is therefore a natural choice for use as an optimizer. Inspired by the single-objective particle swarm optimizer using ring topology proposed by Li [12], this paper proposes a multi-objective particle swarm optimization algorithm with ring topology, and includes a special crowding distance, which the authors denote by the acronym MO_Ring_PSO_SCD. In this section, the MO_Ring_PSO_SCD procedure is presented followed by a discussion of its underlying mechanism for the purpose of analyzing how and why the proposed algorithm can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

successfully address multimodal multi-objective problems. Finally, the novelty of the proposed algorithm is described and the convergence behavior of MO_Ring_PSO_SCD is compared with that of alternative algorithms.

A. Procedure of MO_Ring_PSO_SCD

In MO_Ring_PSO_SCD, the personal best archive (***PBA***) and the neighborhood best archive (***NBA***) are first established, and then the *pbest* and *gbest* are selected from the two respective archives. Ring topology is used to induce multiple niches. In addition, a special selection scheme is proposed to maintain more Pareto-optimal solutions. The procedure of MO_Ring_PSO_SCD is shown in **Algorithm 1**, where $\boldsymbol{P}$ represents the whole population and $\boldsymbol{P}_i(t)$ stands for the i^{th} particle at the t^{th} generation. The personal best positions are retained in ***PBA***, where the notation $\boldsymbol{PBA}\{i\}$ represent the i^{th} paticle's best positions found so far. The ***PBA*** acts as an 'anchor' that provides a stable network for the whole population. Due to the use of the ***PBA***, the position of each particle can be improved generation by generation in a stable fashion.

The neighborhood best positions are represented as ***NBA***, where $\boldsymbol{NBA}\{i\}$ denotes the best positions within the i^{th} particle's neighborhood. There are three particles in each neighborhood and each particle interacts with its immediate neighbors on its right and left. Instead of using the global best positions of the whole population, the neighborhood best positions of each particle are employed to avoid having the population converge to a single point. Since the neighborhood is established using an index-based ring topology, particles in different neighborhood cannot interact with each other directly. The adoption of the ***NBA*** restricts the information transmission through the population, hence allowing the formation of multiple niches during the search.

The general procedure for implementing the MO_Ring_PSO_SCD algorithm is as follows. First, the whole population ($\boldsymbol{P}$), the personal best archive (***PBA***) and the neighborhood best archive (***NBA***) are initialized. Then, the i^{th} particle's leaders (denoted as $pbest_i$ and $nbest_i$) are respectively chosen from $\boldsymbol{PBA}\{i\}$ and $\boldsymbol{NBA}\{i\}$ according to the non-dominated_scd_sort algorithm. After sorting, the first particle in the sorted $\boldsymbol{PBA}\{i\}$ is chosen as $pbest_i$ and the first one in the sorted $\boldsymbol{NBA}\{i\}$ is chosen as $nbest_i$. Then $\boldsymbol{P}_i(t)$ is updated to $\boldsymbol{P}_i(t+1)$ according to (4) and (5). After evaluation, $\boldsymbol{P}_i(t+1)$ is stored into $\boldsymbol{PBA}\{i\}$ and all particles dominated by $\boldsymbol{P}_i(t+1)$ are removed. Then $\boldsymbol{NBA}\{i\}$ (the i^{th} particle's neighborhood best archive) is updated. The neighborhood of the i^{th} particle ($1 < i < ParticleNumber$) includes the $(i-1)^{th}$ particle, the i^{th} particle and the $(i+1)^{th}$ particle. The neighborhood of the first particle contains the last particle, the first particle and the second particle, while the neighborhood of the last particle contains the $(ParticleNumber-1)^{th}$ particle, as well as the last and the first particles. The non-dominated particles in the personal best archives of the i^{th} particle's neighborhood are selected as the updated $\boldsymbol{NBA}\{i\}$. The above steps are repeated until the termination conditions are met. The description of non-dominated_scd_sort and the method to calculate the special crowding distance are presented in the following parts.

Algorithm 1:MO_Ring_PSO_SCD

```
1  //Initialize Population P(0)
2        Evaluation(P(0))
3  //Initialize PBA and NBA
4        for i = 1: ParticleNumber
5          PBA{i} = P_i (0)
6          NBA{i} = PBA{i}
7        end for
8  while Generation < MaxGenerations do
9        for i = 1 : ParticleNumber
10    //Sort particles in PBA and NBA
11       Sorted_PBA{i} = non-dominated_scd_sort(PBA{i})
12       Sorted_NBA{i} = non-dominated_scd_sort(NBA{i})
13    //Select pbest and nbest
14          pbest_i = The first particle in Sorted_PBA{i}
15          nbest_i = The first particle in Sorted_NBA{i}
16          Update P_i(t) to P_i(t+1) according to (4) and (5)
17          Evaluation(P_i(t+1))
18    //Update PBA
19    Put P_i(t+1) into PBA{i} and remove particles dominated
      by P_i(t+1)
20     end for
21    //Update NBA
22       for i = 1: ParticleNumber
23          if i = 1
24          temp_NBA{i} = [PBA{ ParticleNumber },
                           PBA{1}, PBA{2}]
25          else if  i = ParticleNumber
26          temp_NBA{i} = [PBA{ ParticleNumber -1},
                           PBA{ ParticleNumber }, PBA{1}]
27          else
28          temp_NBA{i} = [PBA{i-1}, PBA{i}, PBA{i+1}]
29          end if
30        NBA{i} = non-dominated particles in temp_NBA{i}
31        end for
32 end while
33 Output the non-dominated particles in NBA
```

1) Description of non-dominated_scd_sort

The non-dominated_scd_sort is carried out in two steps. In the first step, the particles are sorted according to non-dominated sorting scheme [23]. Then, in the second step the special crowding distances of non-dominated particles are calculated. The non-dominated solutions are ranked in descending order according to their special crowding distances. After sorting, the first particle is the non-dominated solution with the largest special crowding distance. Details of calculating the special crowding distance are described in the next part.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

2) Special crowding distance

The special crowding distance, SCD, used in the proposed algorithm is a modification of the approach implemented in the Omni-optimizer technique [20]. The modified procedure involves two steps. The first step involves the calculation of the *crowding distance*, CD, for each particle in the decision space and for its corresponding image in the objective space. In the second step, the crowding distances from the first step are used to assign an SCD metric to each particle in the decision and objective spaces, using a criterion described below.

To describe the first step of the SCD calculation procedure, let $CD_{i,x}$ be the crowding distance of particle i in the decision space, $CD_{i,f}$ be the crowding distance of the image of particle i in the objective space, and let $CD_{avg,x}$ and $CD_{avg,f}$ respectively be the average crowding distances in the decision and objective spaces. Fig. 3 illustrates the procedure used to calculate the crowding distances $CD_{i,x}$ in the decision space, for the simple case of five particles and two decision variables x_1 and x_2. The calculation of the crowding distances $CD_{i,f}$ in the objective space is carried out in an analogous fashion to that adopted for the decision space, following the reasoning illustrated with the assistance of Fig. 3(a) and Fig. 3(b), along with an obvious adaptation of the corresponding formula to denote elements in the objective space. However, when considering boundary points in the objective space (*i.e.,* the analog of Fig. 3(c)), the proposed method adopts a new convention. More precisely, to a particle i which is the boundary point in the objective space, we assign the crowding metric as follows. In minimization problems, when particle i has the minimum value for the m^{th} objective, the contribution of the m^{th} objective in $CD_{i,f}$ is set to 1 and when it has the maximum value for the m^{th} objective, the contribution of m^{th} objective in $CD_{i,f}$ is set to 0. Conversely, in maximization problems, when particle i has the minimum value in the m^{th} objective, the contribution of m^{th} objective in $CD_{i,f}$ is set to 0. When particle i has the maximum value in the m^{th} objective, the contribution of m^{th} objective in $CD_{i,f}$ is set to 1. For example, assuming that Fig. 3(c) shows the distribution of five solutions to a minimization problem in the objective space, x_1 and x_2 in Fig. 3(c) are respectively replaced by f_1 and f_2. Since particle 5 has the maximum value for the first objective. The contribution of the first objective in $CD_{5,f}$ is set to 0. However, particle 5 has the minimum value for the second objective. The contribution of the second objective in $CD_{5,f}$ is set to 1. Therefore, the $CD_{5,f} = (0 + 1) = 1$. This boundary-point metric assignment is a significant departure from the method in [20]. In Omni-optimizer, if particle i is the minimum solution in the m^{th} objective, $CD_{i,f}$ is set to ∞. Then the $CD_{avg,f}$ is equal to ∞. In the stage of 'Determine the final crowding distance', there is no chance that $CD_{i,f} > CD_{avg,f}$ since the $CD_{avg,f}$ is equal to ∞.

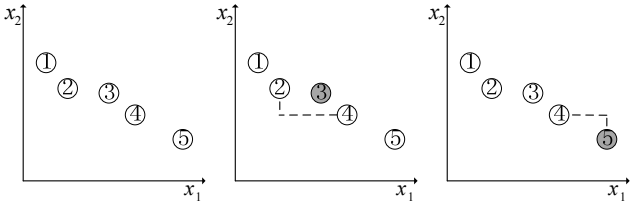
In the second step, the special crowding distance SCD_i for each particle i, is assigned as

$$SCD_i = \max(CD_{i,x}, CD_{i,f}) \quad (6)$$

when either $CD_{i,x} > CD_{avg,x}$ or $CD_{i,f} > CD_{avg,f}$; otherwise we set

$$SCD_i = \min(CD_{i,x}, CD_{i,f}) \quad (7)$$

These equations show that the SCD concept involves a max or min selection step that involves crowding metrics from the decision and objective spaces. This methodology can therefore promote diversity in both spaces simultaneously.



(a) Rank the particles according to x_1;
(b) The crowding distance in decision space of particle 3, $CD_{3,x} = \frac{|x_{1,4} - x_{1,2}|}{|x_{1,5} - x_{1,1}|} + \frac{|x_{2,4} - x_{2,2}|}{|x_{2,5} - x_{2,1}|}$ ($x_{i,j}$ represents the i^{th} dimension value of the j_{th} particle);
(c) The crowding distance in decision space of boundary particle 5, $CD_{5,x} = 2*(\frac{|x_{1,5} - x_{1,4}|}{|x_{1,5} - x_{1,1}|} + \frac{|x_{2,5} - x_{2,4}|}{|x_{2,5} - x_{2,1}|})$; (Note that the crowding distance in objective space of boundary is different from that of decision space.)

Fig. 3. Illustration of the proposed crowding distance calculating method in decision space.

B. Mechanism of MO_Ring_PSO_SCD

The combination of ring topology and the special crowding distance can be used to advantage for solving multimodal multi-objective problems effectively. The reason is that a central characteristic of multimodal multi-objective problems is that there may be two or more Pareto-optimal solutions corresponding to the same point in the objective space. As soon as one of the Pareto-optimal solutions is found, it is more likely to be chosen as a parent or leader for the next generation and the next population will be guided towards it. Therefore, the other Pareto-optimal solutions, which correspond to the same point in the objective space, become difficult to approach. Even though these alternative optimal solutions have been successfully located, it is hard for them to survive in the environmental selection process, because they crowd near the previous solution in the objective space [4]. An algorithm should have two kinds of capabilities to solve multimodal multi-objective problems, namely the ability to (*i*) find as many as possible Pareto-optimal solutions, and (*ii*) maintain the Pareto-optimal solutions that correspond to the same point in the objective space. In MO_Ring_PSO_SCD, with the assistance of ring topology, each particle only exchanges information with its closed neighbors. Thus, niches are formed in the population, and consequently the diversity of population is improved. Therefore, MO_Ring_PSO_SCD is able to search out more Pareto-optimal solutions. In addition, the use of the special crowding distance as a second environmental selection criterion enables MO_Ring_PSO_SCD to maintain the Pareto-optimal solutions that correspond to the same point in the objective space. For example, assume that solutions A_1 and A_2 have been located, as shown in Fig. 2. Their crowding distance in the objective space is zero (because both solutions correspond to A'). However, the distance between the two solutions in the decision space is large. The special crowding distance adaptively chooses the distance in the decision space as their crowding distance according to (6). Therefore, A_1 and

A_2 are both likely to be selected as leaders in different niches of the next generation. Guided by leaders, the particles in niches corresponding to A_1 and A_2 will fly towards them separately. Therefore, both PS_1 and PS_2 are likely to be obtained generation after generation. This provides a reason why the combination of ring topology and special crowding distance, as implemented in MO_Ring_PSO_SCD, leads to a more effective approach to multimodal multi-objective problems. Experimentally verification of the effectiveness of the proposed algorithm is given in Section VI.B.

C. The Novelty and Convergence Behavior of MO_Ring_PSO_SCD

The inspiration for MO_Ring_PSO_SCD is from r3pso, while the special crowding distance is an improved version from that used by the Omni-optimizer technique. The novelty of the MO_Ring_PSO_SCD is perhaps best to put into evidence by comparing and contrasting it with the r3pso and Omni-optimizer techniques and the related DN-NSGAII method.

The differences between MO_Ring_PSO_SCD and r3pso are as follows. First, r3pso is a single objective optimization algorithm. In contrast, MO_Ring_PSO_SCD is a multi-objective optimization technique. Second, the purpose of r3pso is to identify multiple local and global optima for single optimization problems, while MO_Ring_PSO_SCD aims to obtain several PSs (each containing multiple solutions) that correspond to the same PF. Therefore, the number of solutions that MO_Ring_PSO_SCD maintains is much greater than that for r3pso method. Third, r3pso selects the *gbest* and *nbest* only according to fitness values, while in contrast, MO_Ring_PSO_SCD selects the *gbest* and *nbest* according to a dominated relationship and the special crowding distance.

There are numerous differences between MO_Ring_PSO_SCD and the Omni-optimizer method. The Omni-optimizer technique is similar to NSGAII, with a population that evolves by crossover and mutation. In contrast, the particles in MO_Ring_PSO_SCD are led by *pbest* and *gbest*. Both the MO_Ring_PSO_SCD and the Omni-optimizer methods make use of the crowding distance in the decision space and objective space. However, the way MO_Ring_PSO_SCD deals with boundary points in objective space is significantly different from the Omni-optimizer's approach. In addition, the topological structure in MO_Ring_PSO_SCD is fundamentally different from that of the Omni-optimizer.

The differences between MO_Ring_PSO_SCD and DN-NSGAII are presented as follows. (*i*) DN-NSGAII is modified from NSGAII. In DN-NSGAII, the particles are changed by crossover and mutation. However, the framework of MO_Ring_PSO_SCD is multi-objective PSO. The positions of particles are changed according to *pbest* and *gbest*. (*ii*) There is no specific topology in DN-NSGAII. In contrast, ring topology is employed in MO_Ring_PSO_SCD. Each particle transfers information with its immediate neighbors. (*iii*) The crowding distance in DN-NSGAII represents distance in decision space. However MO_Ring_PSO_SCD employs special crowding distance (the special crowding distance makes use of distance in both decision space and objective space).

The MO_Ring_PSO_SCD, Omni-optimizer and DN-NSGAII algorithms are tested on multimodal multi-objective test function 4 (MMF4) to reveal their differences in convergence behavior (note that r3pso is a single objective optimization algorithm, so it cannot be tested on the multimodal multi-objective problems and hence has been excluded from this comparison study). Test function MMF4, whose details are given in Appendix.A, has four PSs in its feasible region with a distribution as shown in Fig. 4. In Fig. 4, x_1 and x_2 represent the two dimensions of the decision space. The feasible region of MMF4 is divided into four subregions, namely Region 1 $\{x_1 \in [-1, 0], x_2 \in [1, 2]\}$, Region 2 $\{x_1 \in (0, 1], x_2 \in [1, 2]\}$, Region 3 $\{x_1 \in [-1, 0], x_2 \in (0, 1]\}$, and Region 4 $\{x_1 \in (0, 1], x_2 \in [0, 1)\}$. There is one PS in each region. The proportion of solutions in each region is calculated for every generation to show the algorithm's convergence behavior. Ideally, if the algorithm performs well on MMF4, the proportion of solutions in each of the four regions should converge to 25%. In this example, the population size of MO_Ring_PSO_SCD, Omni-optimizer and DN-NSGAII is set to 800 and the maximal generation number is set to 100. All other parameters are the same as reported in Section VI.A. Each algorithm is run 20 times and the mean proportions of their solutions in each region from the 1st to the 100th generation are plotted in Fig. 5. The top figure of Fig. 5 shows the convergence behavior of the proposed MO_Ring_PSO_SCD algorithm. As shown, the proportions of solutions in Regions 1 and 3 increase to a small extent while the proportions of Regions 2 and 4 solutions decrease slightly as the generation number reaches approximately 20. At approximately the 50th generation the proportion of solutions in each region approaches 25%. In addition, these proportions remain relatively constant from the 50th to 100th generation. In contrast, the convergence behavior of the Omni-optimizer method, as shown in the lower-left graph of Fig. 5, is inferior to that of MO_Ring_PSO_SCD. In fact, from the 40th to 100th generations the proportions of solutions for the Omni-optimizer approach in Regions 1 and 4 are larger than those in Regions 2 and 3. The proportion of solutions in each region fluctuates frequently throughout generations. The convergence behavior of DN-NSGAII is shown in the lower-right graph of Fig. 5. The proportions of solutions in Regions 2 and 3 are much greater than those of Region 1 and Region 4. In addition, the proportion of solutions in each region is not constant as a function of the generation number.

From the above discussion, it follows that a central novelty of the MO_Ring_PSO_SCD is that the particles are led by *pbest* and *gbest*, where each particle transfers information with its immediate neighbors, while a dominance relationship and the special crowding distance are employed for leader selection. The combination of these operations makes it possible for the MO_Ring_PSO_SCD method to solve multimodal multi-objective problems in an effective fashion.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

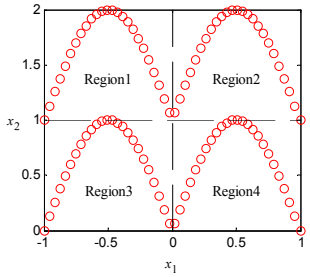


Fig. 4. The distribution of MMF4's PSs.

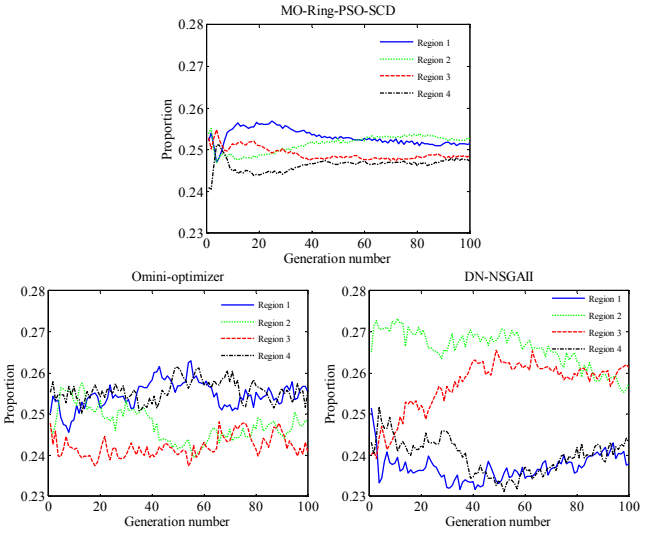


Fig. 5. Convergence behaviors of three algorithms: MO-Ring-PSO-SCD, Omini-optimizer, and DN-NSGAII.

IV. Test Functions

Since the state of the knowledge for solving multimodal multi-objective optimization problems is still in its early stages of development, there are very few benchmark test functions documented in the literature. One of the aims of this paper is to define a set of multimodal multi-objective test functions with the following characteristics:

1) They should be multi-objective optimization functions;
2) They should have more than one PS that corresponds to the same PF;
3) They should vary in their extent of complexity.

A simple way to design a multimodal multi-objective test function is to modify single modal multi-objective benchmark functions. First, the feasible region should be enlarged if it is too small. Then, novel PSs are copied into the new feasible region through shift or symmetry transformation. The complexity of multimodal multi-objective test function relates to the number of different PSs and the overlap among them.

Two simple multimodal multi-objective test functions, namely SS-UF1 and S-UF3, were designed in precious work by the authors [4]. For the details to design multimodal multi-objective test functions please refer to [4]. In this paper, these two multimodal multi-objective test functions (MMF) are renamed as MMF1 and MMF2. In addition, six more complicated test functions MMF3-MMF8 are designed. Three other test functions, called SYM-PART simple [27], SYM-PART rotated [27], and the Omni-test function [20] with $n = 3$, are included in the scope of the study.

Table I shows the most relevant features of the eleven test functions under consideration. The first column of Table I lists the names of all test functions considered in this work. The second column, entitled *Number of PSs* reflects how many PSs correspond to the same PF, and the third column, *Overlap in every dimension*, indicates with a check mark(✓) when the PSs have overlaps in every dimension and with an mark (×) if they do not. Generally speaking, a test function whose PSs overlap in every dimension is more complex than a test function which does not show the same extent of dimensional overlap. In addition, a function is more complex when it has a larger number of PSs.

TABLE I
Features of Test Functions

Function name	Number of PSs	Overlap in every dimension
MMF1	2	×
MMF2	2	×
MMF3	2	✓
MMF4	4	×
MMF5	4	×
MMF6	4	✓
MMF7	2	×
MMF8	4	×
SYM-PART simple	9	✓
SYM-PART rotated	9	✓
Omni-test(n=3)	27	✓

As is shown in Table I, the functions MMF1 and MMF2 have two PSs, and their PSs do not overlap in every dimension. Hence, MMF1 and MMF2 are relatively simple functions. Next, MMF3 has two PSs which overlap in every dimension. Therefore, MMF3 is more complex than MMF1 and MMF2. Function MMF4 and MMF8, which in contrasts to the other functions in the table has a concave PF, have four PSs that do not overlap in every dimension. MMF6 have four PSs which overlap in every dimension. The PSs of MMF7 are irregular curves which are shown in Appendix.A. Function SYM-PART rotated is generated from a simple rotation of SYM-PART simple, and is thus more complex than SYM-PART simple. To make the visualization more accessible, the dimension of the decision space n is set to 3 in the Omni-test, which is indicated in Table I. Omni-test has the largest number of PSs which in addition overlap in every dimension. Therefore, Omni-test is the most complex among the eleven test functions. Further details of the eleven test functions are given in Appendix.A.

V. Performance Indicators

In this paper, the *Pareto Sets Proximity* (*PSP*)

$$PSP = \frac{CR}{IGDX} \tag{8}$$

is proposed as a new indicator to reflect the similarity between the obtained PSs and the true PSs, where *CR* is the *cover rate* and *IGDX* [22] is the *inverted generational distance* in the decision space. The cover rate is a modification of the *maximum spread* (*MS*) [28]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

$$MS = \sqrt{\frac{1}{m}\sum_{l=1}^{m}\delta_l} \tag{9}$$

where

$$\delta_l = \left(\frac{\min(f_l^{\max}, F_l^{\max}) - \max(f_l^{\min}, F_l^{\min})}{F_l^{\max} - F_l^{\min}} \right)^2 \tag{10}$$

and where m is the dimensionality of objective space, $f_l^{\max}$ and $f_l^{\min}$ are respectively the maximum and minimum of obtained PF for the l^{th} objective, $F_l^{\max}$ and $F_l^{\min}$ are the maximum and minimum of the true PF for the l_{th} objective. Note that if $f_l^{\min} \geq F_l^{\max}$, then $\delta_l = 0$. The maximum spread is conceived to reveal how well obtained PF covers the true PF. The larger the *MS* value is, the better the obtained PF covers the true PF. The limiting value $MS = 1$ means the obtained PF covers completely the true PF.

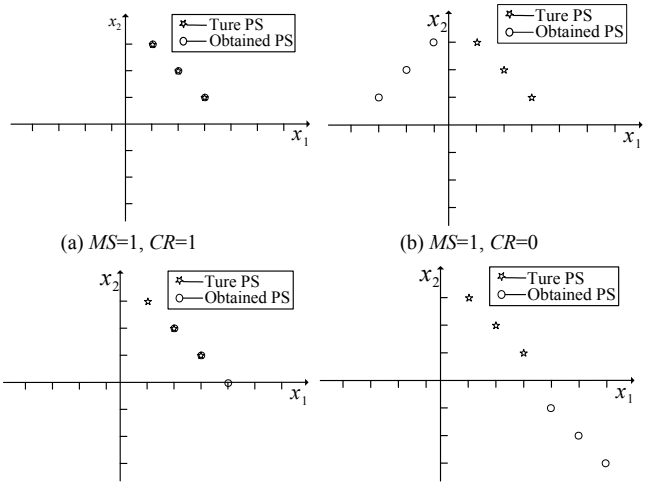
The *MS* indicator has a number of disadvantages. In particular, it ignores the situation where $f_l^{\max} \leq F_l^{\min}$. In addition, the mean value of δ (given by $\frac{1}{m}\sum_{l=1}^{m}\delta_l$) is heavily affected when $\delta_l = 1$, because the maximum of δ is equal to 1. To introduce an indicator into decision space and properly represent overlap ratio between true PS and obtained PS, we introduce Cover Rate (*CR*) defined as:

$$CR = \left(\prod_{l=1}^{n} \delta_l \right)^{1/2n} \tag{11}$$

$$\delta_l = \begin{cases} 1 & V_l^{\max} = V_l^{\min} \\ 0 & v_l^{\min} \geq V_l^{\max} \ \| \ v_l^{\max} \leq V_l^{\min} \\ \left(\dfrac{\min(v_l^{\max}, V_l^{\max}) - \max(v_l^{\min}, V_l^{\min})}{V_l^{\max} - V_l^{\min}} \right)^2 & \text{otherwise} \end{cases} \tag{12}$$

where n is the dimensionality of decision space; $v_l^{\max}$ and $v_l^{\min}$ are respectively the maximum and minimum of obtained PS for the l^{th} variable; $V_l^{\max}$ and $V_l^{\min}$ are the maximum and minimum of the true PS for the l^{th} variable.

Four representative scenarios are shown in Fig. 6 to illustrate the differences in the values of the *MS* and *CR* indicators. It can be concluded in Fig. 6 that, in scenarios (a) and (c), *MS* is equal to *CR*. However, in scenarios (b) and (d), it is obvious that there is no overlap between true PS and obtained PS, but the *MS* values are not equal to 0 while the *CR* adopts the value 0 accurately reflecting the total lack of overlap. It is clear that *CR* is able to adequately address the situation where $f_l^{\max} \leq F_l^{\min}$; furthermore, in contrast to the *MS*, the *CR* remains unaffected by the limiting case where $\delta_l = 1$ .Therefore, the proposed indicator *CR* is more appropriate than *MS*. In conclusion, *CR* can show overlap ratio between true PS and obtained PS. Larger *CR* values are desirable.



(a) *MS*=1, *CR*=1 (b) *MS*=1, *CR*=0

(c) *MS*=0.5, *CR*=0.5 (d) *MS*=0.7, *CR*=0

Fig. 6. Four scenarios for comparing the performance of the *MS* and *CR* indicators.

However, *CR* cannot show the diversity and convergence of the obtained solutions. A new indicator *IGDX* is proposed by Zhou [22] to evaluate the diversity and convergence in the decision space. *IGDX* values represent the average distance (Euclidean distance) between the obtained solutions and reference solutions (true PS) in decision space. Let P^* denote a set of uniformly distributed points along the true PS. Let O denote a set of obtained solutions. The *IGDX* can be calculated as the average distance from P^* to O:

$$IGDX(O, P^*) = \frac{\sum_{v \in P^*} d(v, O)}{|P^*|} \tag{13}$$

where $d(v, O)$ is the minimum Euclidean distance between v and the points in O. Smaller *IGDX* values are desirable.

CR and *IGDX* are combined as *PSP* according to (8). *PSP* can not only reflect the convergence of obtained PS, but also represent overlap ratio between true PS and obtained PS. Larger *PSP* values are desirable.

VI. Experiments and Analysis

A. Experimental Setups

As is well known, different algorithms may favor different population sizes. In addition, the optimal population size is different for different test functions. Therefore, it is impossible to set one population size that is suitable for all algorithms and test functions. In Sections VI.B and VI.C a population size of 800 is taken as an example to put into evidence details of the results produced by the algorithms. Then Section VI.D shows experimental results with different population sizes to properly analyze the effect of population size. For the purpose of unbiased comparison, the maximal number of evaluations is set to 80,000 for all the algorithms. All the experiments are carried out 20 times. In MO_Ring_PSO_SCD, both C_1 and C_2 are set to 2.05 and W is set to 0.7298 in (5). Other parameters are set as the corresponding references [4][20][23] [29][30].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

B. Experimental Verification of the Effectiveness of the Proposed Algorithm

To demonstrate the effectiveness of ring topology and of the special crowding distance, multi-objective PSO algorithms with and without ring topology or special crowding distance are tested on all the test functions. The *PSP* values of the four algorithms are analyzed by rank sum test. The *PSP* values and *h* values of rank sum test are shown in Table II. The Pareto-optimal solutions obtained by multi-objective PSO algorithms on MMF3 are shown in Fig. 7. In Table I and Fig. 7, MO_PSO represents simple multi-objective PSO without ring topology and special crowding distance. MO_Ring_PSO is multi-objective PSO with only ring topology. MO_PSO_SCD is multi-objective PSO with only special crowding distance. MO_Ring_PSO_SCD is multi-objective PSO with both ring topology and special crowding distance. It can be concluded from Table II and Fig. 7 that MO_Ring_PSO is much better than MO_PSO, while MO_PSO_SCD is a little better than MO_PSO. In short, the inclusion of both ring topology and special crowding distance are effective for improving performance. In addition, MO_Ring_PSO_SCD is the best performer over all the eleven test functions. The rank sum test results show that there are significant differences among MO_Ring_PSO_SCD and the other three algorithms. Why these two operates are effective in solving multimodal multi-objective optimization? The reasons are analyzed in the following paragraphs.

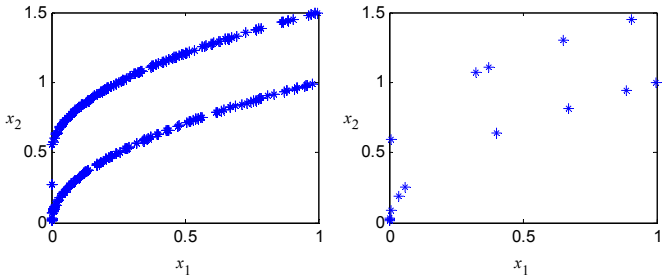
Ring topology enables the proposed algorithm to locate enough Pareto-optimal solutions. It is proved that ring topology induces stable niches in decision space [12]. Therefore, the particles evolve within their own niches. Every niche has its own leader. If the leaders are well-distributed in the decision space, it is more likely to locate more Pareto-optimal solutions.

The introducing of special crowding distance improves the population diversity and helps maintain more Pareto-optimal solutions. The less crowded solutions are preferred. Therefore, the diversity of obtained solution is improved. In environmental selection, if solutions are crowded in objective space but far away from each other in decision space, they are able to survive according to their special crowding distances.

In conclusion, both ring topology and special crowding distance make MO_Ring_PSO_SCD more excellent in solving multimodal multi-objective problems.

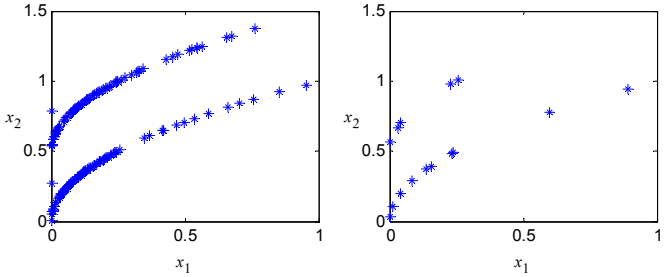
TABLE II
PSP VALUES OF MULTI-OBJECTIVE PSO WITH AND WITHOUT RING TOPOLOGY OR SPECIAL CROWDING DISTANCE

	MO_Ring_PSO_SCD mean ± std dev	MO_PSO_SCD mean ± std dev (*h*)	MO_Ring_PSO mean ± std dev (*h*)	MO_PSO mean ± std dev (*h*)
MMF1	**66.80** ± 2.89	3.25 ± 0.35 (1)	42.17 ± 2.51 (1)	3.05 ± 0.73 (1)
MMF2	**98.13** ± 12.91	8.92 ± 2.20 (1)	22.82 ± 7.19 (1)	4.01 ± 1.88 (1)
MMF3	**132.71** ± 23.13	6.87 ± 1.57 (1)	34.42 ± 9.45 (1)	5.60 ± 1.99 (1)
MMF4	**115.30** ± 4.33	1.61 ± 0.32 (1)	43.54 ± 3.53 (1)	1.67 ± 0.38 (1)
MMF5	**32.91** ± 1.27	2.19 ± 0.33 (1)	24.05 ± 0.94 (1)	2.13 ± 0.44 (1)
MMF6	**36.01** ± 1.26	2.48 ± 0.27 (1)	27.39 ± 1.25 (1)	2.28 ± 0.41 (1)
MMF7	**108.01** ± 5.88	3.59 ± 0.44 (1)	78.58 ± 2.92 (1)	4.23 ± 0.80 (1)
MMF8	**47.21** ± 1.42	1.06 ± 0.24 (1)	32.51 ± 1.98 (1)	1.45 ± 0.33 (1)
SYM-PART simple	**21.46** ± 1.01	0.50 ± 0.14 (1)	15.26 ± 0.94 (1)	0.56 ± 0.41 (1)
SYM-PART rotated	**18.27** ± 1.52	1.06 ± 0.44 (1)	14.78 ± 0.82 (1)	0.47 ± 0.31 (1)
Omni-test	**11.45** ± 0.58	0.78 ± 0.09 (1)	5.68 ± 1.40 (1)	0.69 ± 0.07 (1)



(a) PSs obtained by MO_Ring_PSO_SCD (b) PSs obtained by MO_PSO_SCD

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation



(c) PSs obtained by MO_Ring_PSO (d) PSs obtained by MO_PSO

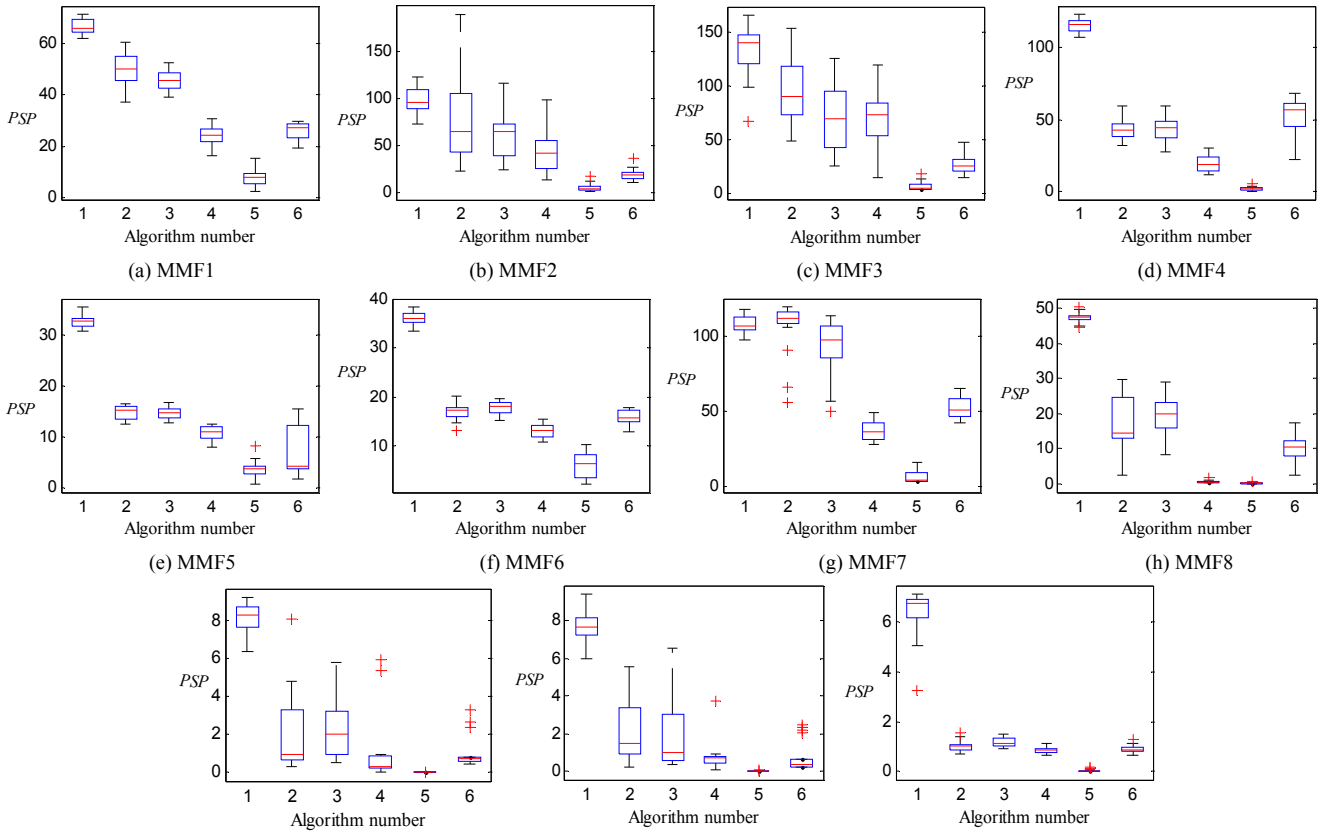
Fig. 7. The comparison of PSs obtained by multi-objective PSO algorithms on MMF3.

C. Comparison with Other Algorithms

MO_Ring_PSO_SCD is compared with five algorithms on eleven test functions. Two measure indicators, *PSP* and *Hv* are adopted to compare the performances of different algorithms. *PSP* reflects the quality of the obtained PSs in decision space and *Hv* reflects the quality of obtained PF in objective space. Larger *Hv* and *PSP* are more favorable. The statistical results of *PSP* values are shown in Fig. 8 by box-plots. The *Hv* values of different algorithms are shown in Table III.

In Fig. 8, MO_Ring_PSO_SCD, Omni-optimizer, DN-NSGAII, NSGAII, MOEAD and SPEA2 are numbered 1, 2, 3, 4, 5, and 6 respectively. As can be verified from the results, the mean *PSP* values for MO_Ring_PSO_SCD are highest on all test functions except MMF7 since the PSs of MMF7 are in an irregular geometry. Omni-optimizer ranks second on MMF1 and MMF3. On the other test functions, the performances of Omni-optimizer and DN-NSGAII are similar. These two algorithms are a somewhat better than NSGAII, MOEAD and SPEA2, because that both Omni-optimizer and DN-NSGAII employ the crowding distance in the decision space for environmental selection. However, the *PSP* values of Omni-optimizer and DN-NSGAII are not stable, because their variances are relatively large.

The reason why NSGAII, MOEAD and SPEA2 perform poorly on all test functions is that these methods consider only the crowding distance in the objective space. The *PSP* values of all algorithms are lowest on Omni-test among all the test functions, which can demonstrate that Omni-test is the most complex test function among them.



(a) MMF1 (b) MMF2 (c) MMF3 (d) MMF4

(e) MMF5 (f) MMF6 (g) MMF7 (h) MMF8

(i) SYM-PART simple (j) SYM-PART rotated (k) Omni-test

Fig. 8. The box-plots of *PSP* values of different algorithms on eleven test functions. The numerals on the horizontal axis of each plot indicate the following algorithms: 1 = MO_Ring_PSO_SCD, 2 = Omni-optimizer, 3 = DN-NSGAII, 4 = NSGAII, 5 = MOEAD, and 6 = SPEA2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation

TABLE III
Hv VALUES OF DIFFERENT ALGORITHMS

	MO_Ring_PSO_SCD	Omni-optimizer	DN-NSGAII	NSGAII	MOEAD	SPEA2
	mean ± std dev	mean ± std dev	mean ± std dev	mean ± std dev	mean ± std dev	mean ± std dev
MMF1	3.66 ± 4.54e-04	3.67 ± 2.43e-05	3.66 ± 1.12e-03	**3.67 ± 1.26e-03**	3.67 ± 4.93e-04	3.66 ± 3.09e-04
MMF2	3.65 ± 7.61e-03	**3.67 ± 4.69e-05**	3.66 ± 9.08e-03	3.66 ± 4.82e-03	3.67 ± 6.26e-04	3.62 ± 3.49e-02
MMF3	3.65 ± 6.63e-03	**3.67 ± 2.52e-05**	3.67 ± 5.25e-04	3.66 ± 1.14e-02	3.67 ± 3.62e-04	3.64 ± 6.33e-03
MMF4	3.30 ± 9.54e-04	3.32 ± 3.84e-05	3.18 ± 3.31e-04	3.32 ± 2.18e-06	**3.33 ± 2.75e-06**	3.30 ± 3.71e-03
MMF5	3.66 ± 3.89e-04	3.67 ± 1.81e-05	3.67 ± 3.22e-04	**3.67 ± 1.69e-05**	3.67 ± 6.34e-04	3.66 ± 9.62e-04
MMF6	3.66 ± 3.33e-04	**3.67 ± 2.19e-05**	3.66 ± 1.41e-03	3.67 ± 8.45e-05	3.67 ± 4.36e-04	3.66 ± 1.77e-04
MMF7	3.67 ± 2.10e-04	3.67 ± 4.44e-05	3.66 ± 1.24e-03	**3.67 ± 1.73e-05**	3.67 ± 2.46e-04	3.66 ± 8.05e-04
MMF8	3.21 ± 1.09e-03	3.21 ± 1.20e-04	3.21 ± 1.22e-03	**3.21 ± 7.70e-05**	3.21 ± 2.93e-04	3.21 ± 2.38e-03
SYM-PART simple	1.30 ± 1.65e-03	1.32 ± 3.21e-04	1.32 ± 1.94e-04	1.32 ± 2.32e-04	**1.32 ± 8.66e-05**	1.32 ± 5.25e-04
SYM-PART rotated	1.29 ± 3.55e-03	1.32 ± 2.55e-04	**1.32 ± 4.49e-04**	1.32 ± 2.41e-04	1.32 ± 9.68e-04	1.32 ± 1.09e-02
Omni-test	61.93 ± 2.14e-01	62.06 ± 2.46e-04	62.06 ± 3.95e-04	**62.06 ± 1.20e-04**	62.06 ± 5.07e-03	61.94 ± 4.09e-02

In Table III, the *Hv* values of Omni-optimizer on MMF2, MMF3 and MMF6 are highest among the algorithms. NSGAII obtains the highest *Hv* values on MMF1, MMF5, MMF7, MMF8 and Omni-test functions. MOEAD is the best on MMF4 and SYM-PART simple test functions. DN-NSGAII obtains the best *Hv* value on SYM-PART rotated test function. MO_Ring_PSO_SCD is comparable with all the other algorithms on all the eleven test functions, though it is not the best. In fact, the *Hv* values for all the algorithms are close to each other. The reason is that all the algorithms consider the distribution in the objective space.

Two typical test functions SYM-PART simple and Omni-test are chosen to reveal the obtained PSs and PF for the algorithms. The PSs obtained by different algorithms on SYM_PART simple and Omni-test are shown in Fig. A12 and Fig. A13 in Appendix.B. It is obvious that MO_Ring_PSO_SCD obtains the largest number of Pareto-optimal solutions. The SYM-PART simple test function has nine PSs. Omni-optimizer and DN-NSGAII obtain five of them (several of the obtained PSs are not complete). MOEAD obtains only one PS. A similar situation arises with the Omni-test function, as shown in Fig. A12, which illustrates the fact that it is difficult to locate and maintain several PSs simultaneously without niching operation. The PFs obtained by different algorithms on SYM-PART simple and Omni-test are shown in Fig. A14 and Fig. A15 of Appendix.B. From these figures, it is concluded that all of the six algorithms obtain well-distributed PF on the SYM-PART simple test function. Note that under the Omni-test, the PF of the MO_Ring_PSO_SCD method is not as good as the PFs of Omni-optimizer, DN-NSGAII, and of NSGAII.

In conclusion, MO_Ring_PSO_SCD obtains the best distribution in decision space on all the eleven test problems except MMF7. Though MO_Ring_PSO_SCD doesn't obtain the best distribution in objective space, it is competitive with that of the other algorithms.

D. Experimental Results with Different Population Sizes

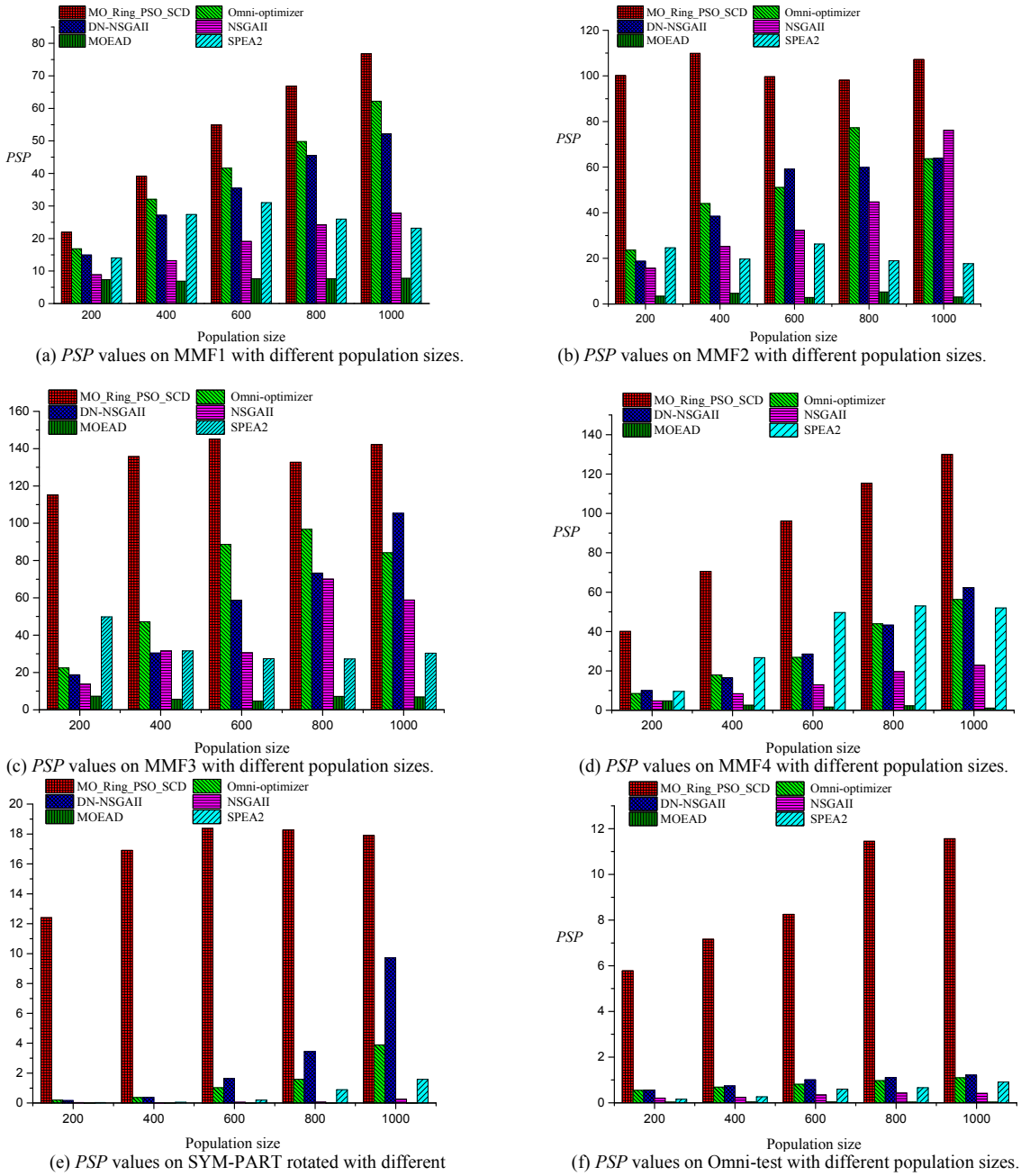
The population size affects the performance of most algorithms. In this subsection, different population sizes are investigated for six of the test functions. The mean *PSP* values for a total of 20 experimental runs are presented in Fig. 9(a)-(f). The figures show that MO_Ring_PSO_SCD achieves the largest *PSP* values under different population sizes for the test functions under consideration. The results validate the claim that the proposed method is effective in solving multimodal multi-objective problems.

The results reported in Fig. 9 also confirm that, as expected, the performance of the algorithms is affected by varying population size. However, the optimal population size for each algorithm varies for the different test functions. With respect to MMF1, the Fig. 9(a) shows that the *PSP* values of all algorithms increases along with the population size, except for the SPEA2 and MOEAD methods. The results obtained with MO_Ring_PSO_SCD do not change much on MMF2, and the *PSP* value of MO_Ring_PSO_SCD is largest when the population size is 400. In contrast, Omni-optimizer achieves its largest *PSP* values when the population size is 800. As for MMF3 both Omni-optimizer and NSGAII perform best with a population size of 800. SPEA2 performs better than Omni-optimizer, DN-NSGAII, NSGAII and MOEAD on MMF4. Therefore, SPEA2 is relatively suitable for problems of the kind represented by MMF4. Finally, from Fig. 9(e)-(f) it is noted that, with the exception of MO_Ring_PSO_SCD, which displays consistently good performance, all the other algorithms perform poorly on the relatively complex functions.

VII. CONCLUSIONS

To solve multimodal multi-objective problems, a multi-objective particle swarm optimization using ring topology and special crowding distance is proposed. The ring topology helps to induce stable niches, so that much more Pareto-optimal solutions can be located. In addition, the special crowding distance considers the crowding distance both in decision and objective space to maintain multiple PSs. The proposed algorithm is compared with five algorithms, three of which are state-of-the-art multi-objective algorithms and two are multimodal multi-objective algorithms. All the algorithms are tested on eleven multimodal multi-objective test functions. Results show that the proposed algorithm is superior to the others in decision space distribution. However, MO_Ring_PSO_SCD does not perform best on multimodal multi-objective test functions with irregular PSs, such as MMF7. In addition, Omni-optimizer and NSGAII are better than the proposed algorithm in the objective space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2754271, IEEE Transactions on Evolutionary Computation



(a) *PSP* values on MMF1 with different population sizes.

(b) *PSP* values on MMF2 with different population sizes.

(c) *PSP* values on MMF3 with different population sizes.

(d) *PSP* values on MMF4 with different population sizes.

(e) *PSP* values on SYM-PART rotated with different population sizes.

(f) *PSP* values on Omni-test with different population sizes.

Fig. 9. *PSP* values with different population sizes.

In our future work, new multimodal multi-objective test functions of different complexity will be proposed. Moreover, the MO_Ring_PSO_SCD will be improved and it will be applied to real world applications. Only problems with multiple global PSs are included in this study. Arguable, there must be problems with multiple local PSs which have been excluded from the scope of the work and that should be a relevant topic for future research.

An intriguing approach to evaluate is the inclusion of multimodal many-objective optimization test functions, for which the ε-dominance distance approach in [31] has performed to advantage. The algorithm proposed by the authors outperforms the ε-dominance distance and the crowding distance in NSGAIIr [32], though details are omitted in this paper for brevity. Future work should include other varieties of test functions to develop a more comprehensive comparative performance analysis for multimodal many-objective optimization problems.

ACKNOWLEDGMENT

The authors are grateful to Prof. Oscar D Crisalle of the University of Florida for valuable editorial comments.

REFERENCES

[1] O. M. Shir, M. Preuss, and B. Naujoks, "Enhancing decision space diversity in evolutionary multiobjective algorithms," in *Proc.*

International Conference on Evolutionary Multi-Criterion Optimization. 2009, pp. 95-109.

[2] M. Preuss, C. Kausch, and C. Bouvy, "Decision space diversity can be essential for solving multiobjective real-world problems," *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, 2010, pp. 367-377.

[3] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "A study on performance metrics to identify solutions of interest from a trade-off set," in *Proc. Australasian Conference on Artificial Life and Computational Intelligence*, 2016, pp. 66-77.

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

[5] B. Y. Qu and P. N. Suganthan, "Novel multimodal problems and differential evolution with ensemble of restricted tournament selection," in *Proc. IEEE Congress on Evolutionary Computation*, 2010, pp. 3480-3486.

[6] M. Preuss, "Niching methods and multimodal optimization performance," in *Proc. Multimodal Optimization by Means of Evolutionary Algorithms*, 2015, pp. 115-137.

[7] J. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, 1975.

[8] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. The Second International Conference on Genetic Algorithms*, 1987, pp. 41-49.

[9] O. J. Mengshoel and D. E. Goldberg, "Probabilistic crowding: Deterministic crowding with probabilistic replacement," in *Proc. Genetic and Evolutionary Computation Conference*, 1999, pp. 409-416.

[10] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Congress on Evolutionary Computation*, 1996, pp. 798-803.

[11] J. P. Li, M. E. Balazs, and G. T. Parks, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 207-234, 2002.

[12] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no.1, pp. 150-169, 2010.

[13] R. C. Eberhart, Y. Shi, and J. Kennedy, "Swarm intelligence," *Morgan Kaufmann*, 2001.

[14] J. Kennedy, "Particle swarm optimization," *Encyclopedia of Machine Learning*, 2011, pp. 760-766.

[15] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308, 2006.

[16] S. Lalwani, S. Singhal, and R. Kumar, "A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm," *Transactions on Combinatorics*, vol. 2, no.1, pp. 39-101, 2013.

[17] R. Brits, A. P. Engelbrecht, and F. Van den Bergh, "A niching particle swarm optimizer," in *Proc. The 4th Asia-Pacific Conference on Simulated Evolution and Learning*, 2002, pp. 692-696.

[18] X. Li, "Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization," in *Proc. Genetic and Evolutionary Computation Conference*, 2004, pp. 105-116.

[19] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences*, vol. 197, pp. 131-143, 2012.

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

[21] K. P. Chan and T. Ray, "An evolutionary algorithm to maintain diversity in the parametric and the objective space," in *Proc. International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005, pp. 13-16.

[22] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167-1189, 2009.

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

[24] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[25] A. R. Jordehi, "Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems," *Applied Soft Computing*, vol. 26, pp. 401-417, 2015.

[26] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congress on Evolutionary Computation*, 2002, pp. 1671-1676.

[27] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of EMOA to detect and preserve equivalent Pareto subsets," in *Proc. International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 36-50.

[28] L. Tang and X. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 20-45, 2013.

[29] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[30] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95-100.

[31] M. Koppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proc. The 4th International Conference on Evolutionary Multi-criterion Optimization*, 2007, pp. 727-741.

[32] F. A. Fortin and M. Parizeau, "Revisiting the NSGA-II crowding-distance computation," in *Proc. The 15th Annual Conference on Genetic and Evolutionary Computation*, 2013, pp. 623-630.

Caitong Yue received the B.E. degree from Zhengzhou University, Zhengzhou, China, in 2014. He is pursuing the Ph.D. degree in Zhengzhou University. His current research interests include evolutionary computation, swarm intelligence, multi-objective optimization and sparse optimization.

Boyang Qu received the B.E. degree and Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is an Associate Professor in the School of Electric and Information Engineering, Zhongyuan University of Technology, China. His research interests include machine learning, neural network, genetic and evolutionary algorithms, swarm intelligence, and multi-objective optimization.

Jing Liang received the B.E. degree from Harbin Institute of Technology, China and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. She is currently a Professor in the School of Electrical Engineering, Zhengzhou University, China. Her main research interests are evolutionary computation, swarm intelligence, multi-objective optimization and neural network.

APPENDIX

A. *Details of test functions*

MMF1 [4]

$$\begin{cases} f_1 = |x_1 - 2| \\ f_2 = 1 - \sqrt{|x_1 - 2|} + 2(x_2 - \sin(6\pi|x_1 - 2| + \pi))^2 \end{cases} \quad \text{(A1)}$$

where $1 \le x_1 \le 3$, $-1 \le x_2 \le 1$.

Its true PS is

$$\begin{cases} x_1 = x_1 \\ x_2 = \sin(6\pi|x_1 - 2| + \pi) \end{cases} \quad \text{(A2)}$$

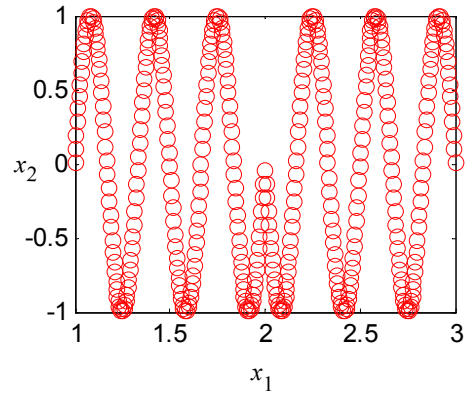
where $1 \le x_1 \le 3$.

Its true PF is

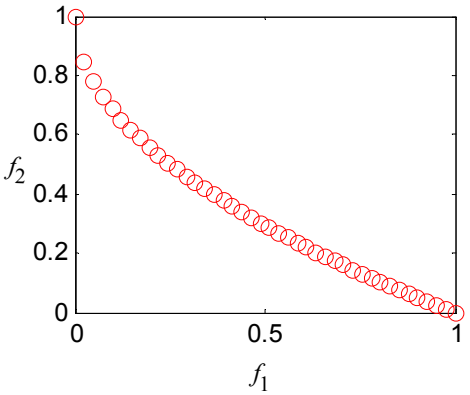
$$f_2 = 1 - \sqrt{f_1} \quad \text{(A3)}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A1.



(a) True PSs of MMF1



(b)True PF of MMF1

Fig. A1. Illustration of the true PSs and PF of MMF1.

MMF2 [4]

$$\begin{cases} f_1 = x_1 \\ f_2 = \begin{cases} 1 - \sqrt{x_1} + 2(4(x_2 - \sqrt{x_1})^2 \\ -2\cos(\dfrac{20(x_2 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2), \quad 0 \le x_2 \le 1 \\ \\ 1 - \sqrt{x_1} + 2(4(x_2 - 1 - \sqrt{x_1})^2 \\ -\cos(\dfrac{20(x_2 - 1 - \sqrt{x_1})\pi}{\sqrt{2}}) + 2), 1 < x_2 \le 2 \end{cases} \end{cases} \quad \text{(A4)}$$

where $0 \le x_1 \le 1$, $0 \le x_2 \le 2$.

Its true PS is

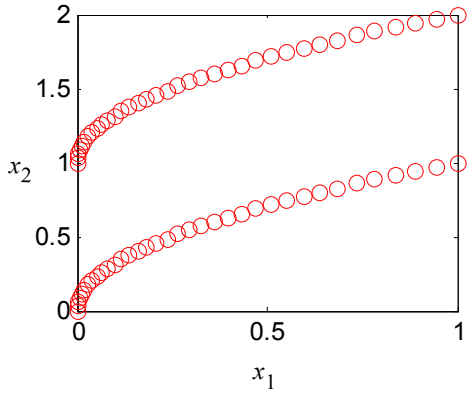
$$\begin{cases} x_2 = x_2 \\ x_1 = \begin{cases} x_2^2 & 0 \le x_2 \le 1 \\ (x_2 - 1)^2 & 1 < x_2 \le 2 \end{cases} \end{cases} \quad \text{(A5)}$$

Its true PF is

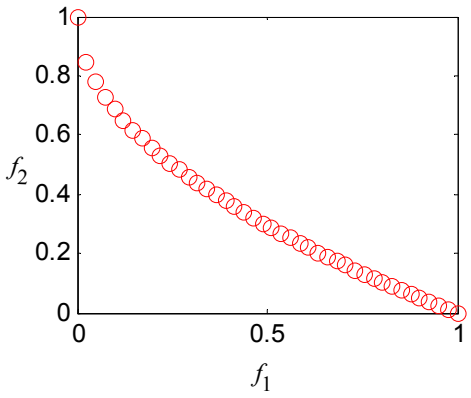
$$f_2 = 1 - \sqrt{f_1} \quad \text{(A6)}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A2.



(a) True PSs of MMF2



(b)True PF of MMF2

Fig. A2. Illustration of the true PSs and PF of MMF2.

MMF3

$$\begin{cases} f_1 = x_1 \\ f_2 = \begin{cases} 1-\sqrt{x_1}+2(4(x_2-\sqrt{x_1})^2- \\ 2\cos(\dfrac{20(x_2-\sqrt{x_1})\pi}{\sqrt{2}})+2) \\ 0 \le x_2 \le 0.5, 0.5 < x_2 < 1 \,\&\, 0.25 < x_1 \le 1 \\ 1-\sqrt{x_1}+2(4(x_2-0.5-\sqrt{x_1})^2 \\ -\cos(\dfrac{20(x_2-0.5-\sqrt{x_1})\pi}{\sqrt{2}})+2) \\ 1 \le x_2 \le 1.5, 0 \le x_1 < 0.25 \,\&\, 0.5 < x_2 < 1 \end{cases} \end{cases} \quad \text{(A7)}$$

where $0 \le x_1 \le 1$, $0 \le x_2 \le 1.5$.

Its true PS is

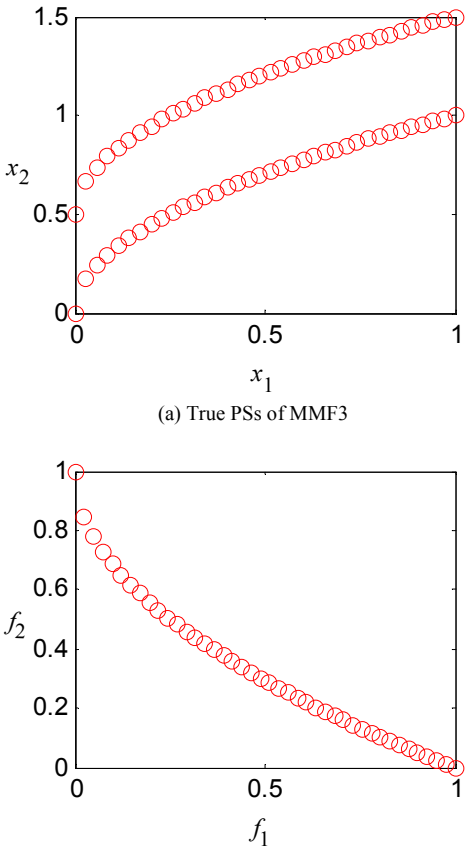
$$\begin{cases} x_2 = x_2 \\ x_1 = \begin{cases} x_2^2 & 0 \le x_2 \le 0.5, 0.5 < x_2 < 1 \,\&\, 0.25 < x_1 \le 1 \\ (x_2-0.5)^2 & 1 \le x_2 \le 1.5, 0 \le x_1 < 0.25 \,\&\, 0.5 < x_2 < 1 \end{cases} \end{cases} \quad \text{(A8)}$$

Its true PF is

$$f_2 = 1-\sqrt{f_1} \quad \text{(A9)}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A3.



(a) True PSs of MMF3

(b)True PF of MMF3

Fig. A3. Illustration of the true PSs and PF of MMF3.

MMF4

$$\begin{cases} f_1 = |x_1| \\ f_2 = \begin{cases} 1-x_1^2+2(x_2-\sin(\pi|x_1|))^2 & 0 \le x_2 < 1 \\ 1-x_1^2+2(x_2-1-\sin(\pi|x_1|))^2 & 1 \le x_2 \le 2 \end{cases} \end{cases} \quad \text{(A10)}$$

where $-1 \le x_1 \le 1$, $0 \le x_2 \le 2$.

Its true PS is

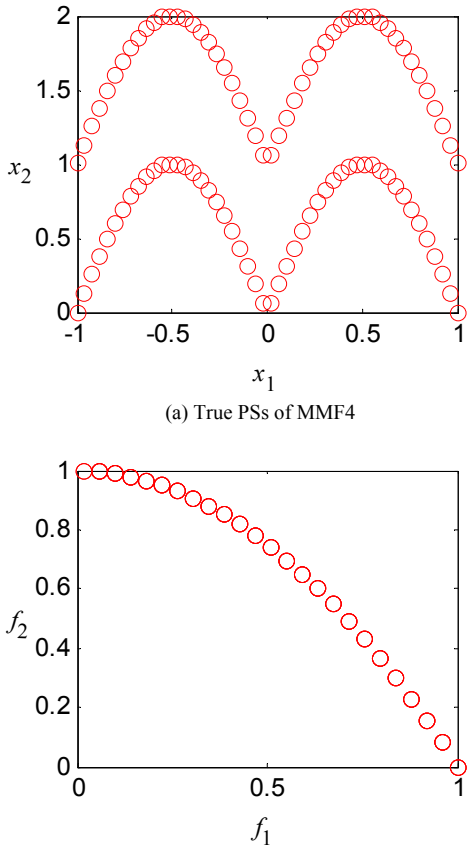
$$\begin{cases} x_1 = x_1 \\ x_2 = \begin{cases} \sin(\pi|x_1|) & 0 \le x_2 \le 1 \\ \sin(\pi|x_1|)+1 & 1 < x_2 \le 2 \end{cases} \end{cases} \quad \text{(A11)}$$

Its true PF is

$$f_2 = 1-f_1^2 \quad \text{(A12)}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A4.



(a) True PSs of MMF4

(b)True PF of MMF4

Fig. A4. Illustration of the true PSs and PF of MMF4.

MMF5

$$\begin{cases} f_1 = |x_1 - 2| \\ f_2 = \begin{cases} 1 - \sqrt{|x_1 - 2|} + 2(x_2 - \sin(6\pi |x_1 - 2| + \pi))^2 & -1 \le x_2 \le 1 \\ 1 - \sqrt{|x_1 - 2|} + 2(x_2 - 2 - \sin(6\pi |x_1 - 2| + \pi))^2 & 1 < x_2 \le 3 \end{cases} \end{cases} \tag{A13}$$

where $-1 \le x_1 \le 3$, $1 \le x_2 \le 3$.

Its true PS is

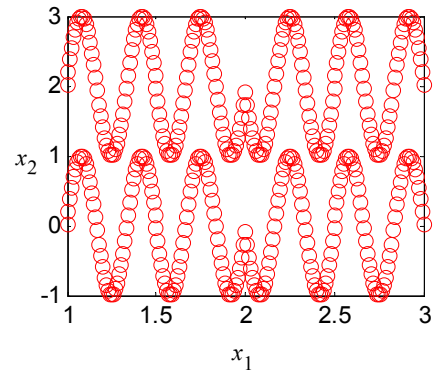
$$x_2 = \begin{cases} \sin(6\pi \,|\, x_1 - 2 \,|\, + \pi) & -1 \le x_2 \le 1 \\ \sin(6\pi \,|\, x_1 - 2 \,|\, + \pi) + 2 & 1 < x_2 \le 3 \end{cases} \tag{A14}$$

Its true PF is

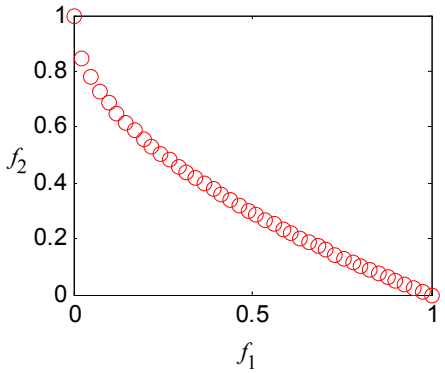
$$f_2 = 1 - \sqrt{f_1} \tag{A15}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A5.



(a) True PSs of MMF5



(b)True PF of MMF5

Fig. A5. Illustration of the true PSs and PF of MMF5.

MMF6

$$\begin{cases} f_1 = |x_1 - 2| \\ f_2 = \begin{cases} 1 - \sqrt{|x_1 - 2|} + 2(x_2 - \sin(6\pi |x_1 - 2| + \pi))^2 & -1 \le x_2 \le 1 \\ 1 - \sqrt{|x_1 - 2|} + 2(x_2 - 1 - \sin(6\pi |x_1 - 2| + \pi))^2 & 1 < x_2 \le 3 \end{cases} \end{cases} \tag{A16}$$

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

Its true PS is

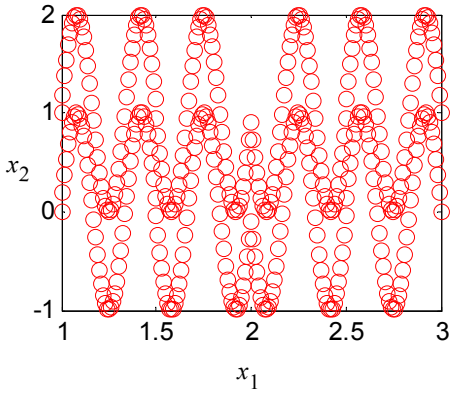
$$x_2 = \begin{cases} \sin(6\pi \,|\, x_1 - 2 \,|\, + \pi) & -1 \le x_2 \le 1 \\ \sin(6\pi \,|\, x_1 - 2 \,|\, + \pi) + 1 & 1 < x_2 \le 2 \end{cases} \tag{A17}$$

Its true PF is

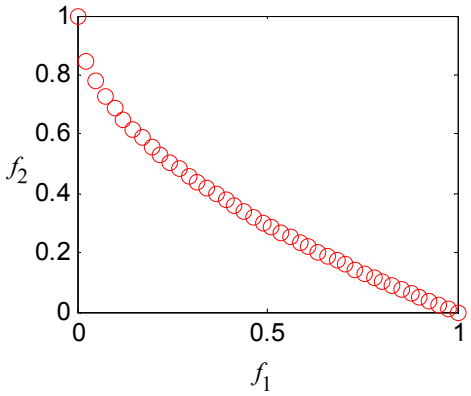
$$f_2 = 1 - \sqrt{f_1} \tag{A18}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A6.



(a) True PSs of MMF6



(b)True PF of MMF6

Fig. A6. Illustration of the true PSs and PF of MMF6.

MMF7

$$\begin{cases} f_1 = |x_1 - 2| \\ f_2 = 1 - \sqrt{|x_1 - 2|} + \{x_2 - [0.3|x_1 - 2|^2 \cdot \cos(24\pi |x_1 - 2| \\ \quad +4\pi) + 0.6|x_1 - 2|] \cdot \sin(6\pi |x_1 - 2| + \pi)\}^2 \end{cases} \tag{A19}$$

where $1 \le x_1 \le 3$, $-1 \le x_2 \le 1$.

Its true PS is

$$x_2 = [0.3|x_1 - 2|^2 \cos(24\pi |x_1 - 2| + 4\pi) + 0.6|x_1 - 2|] \cdot \sin(6\pi |x_1 - 2| + \pi) \tag{A20}$$

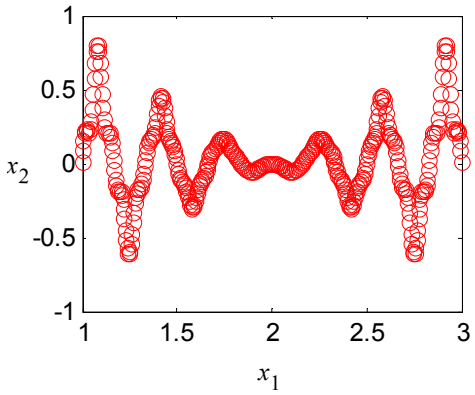
where $1 \le x_1 \le 3$.

Its true PF is

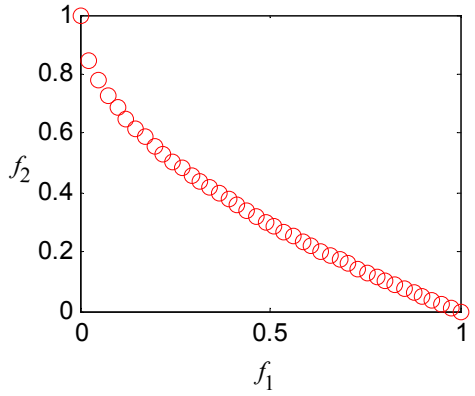
$$f_2 = 1 - \sqrt{f_1} \tag{A21}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A7.

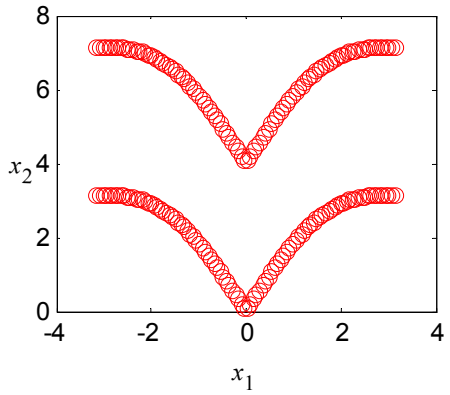


(a) True PSs of MMF7

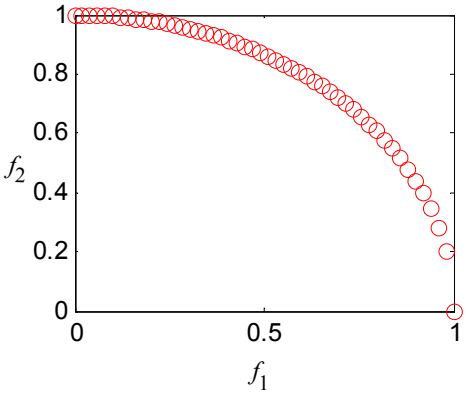


(b)True PF of MMF7

Fig. A7. Illustration of the true PSs and PF of MMF7.



(a) True PSs of MMF8



(b)True PF of MMF8

Fig. A8. Illustration of the true PSs and PF of MMF8.

MMF8

$$\begin{cases} f_1 = \sin|x_1| \\ f_2 = \begin{cases} \sqrt{1-(\sin|x_1|)^2} + 2(x_2 - \sin|x_1| - |x_1|)^2 & 0 \le x_2 \le 4 \\ \sqrt{1-(\sin|x_1|)^2} + 2(x_2 - 4 - \sin|x_1| - |x_1|)^2 & 4 < x_2 \le 9 \end{cases} \end{cases} \tag{A22}$$

where $-\pi \le x_1 \le \pi$, $0 \le x_2 \le 9$.

Its true PS is

$$x_2 = \begin{cases} \sin|x_1| + |x_1| & 0 \le x_2 \le 4 \\ \sin|x_1| + |x_1| + 4 & 4 < x_2 \le 9 \end{cases} \tag{A23}$$

where $-\pi \le x_1 \le \pi$.

Its true PF is

$$f_2 = \sqrt{1 - f_1^2} \tag{A24}$$

where $0 \le f_1 \le 1$.

Its true PS and PF are illustrated in Fig. A8.

SYM-PART simple [27]

$$\begin{cases} \hat{t}_1 = \text{sgn}(x_1) \times \left\lceil \dfrac{|x_1| - (a + \frac{c}{2})}{2a + c} \right\rceil \\ \hat{t}_2 = \text{sgn}(x_2) \times \left\lceil \dfrac{|x_2| - \frac{b}{2}}{b} \right\rceil \end{cases} \tag{A25}$$

$$t_i = \text{sgn}(\hat{t}_i) \times \min\{|\hat{t}_i|, 1\} \tag{A26}$$

$$\begin{cases} p_1 = x_1 - t_1(c + 2a) \\ p_2 = x_2 - t_2 b \end{cases} \tag{A27}$$

$$\begin{cases} f_1 = (p_1 + a)^2 + p_2^2 \\ f_2 = (p_1 - a)^2 + p_2^2 \end{cases} \tag{A28}$$

where $x_i \in [-20, 20]$.

Its true PS is

$$\begin{cases} x_1 = p_1 \\ x_2 = 0 \end{cases} \tag{A29}$$

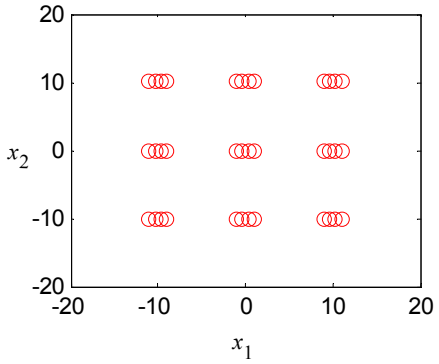
Its true PF is

$$\begin{cases} f_1 = 4a^2 v^2 \\ f_2 = 4a^2 (1 - v)^2 \end{cases} \tag{A30}$$

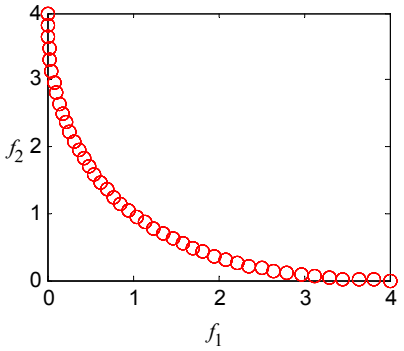
where $v \in [0,1]$.

In this paper, $a=1$, $b=10$, $c=8$.

Its true PS and PF are illustrated in Fig. A9.



(a) True PSs of SYM-PART simple



(b)True PF of SYM-PART simple

Fig. A9. Illustration of the true PSs and PF of SYM-PART simple.

SYM-PART rotated [27]

$$\begin{cases} r_1 = (\cos\omega)\times x_1 - (\sin\omega)\times x_2 \\ r_2 = (\sin\omega)\times x_1 + (\cos\omega)\times x_2 \end{cases} \tag{A31}$$

$$\begin{cases} \hat{t}_1 = \mathrm{sgn}(r_1)\times \left\lceil \dfrac{|r_1| - (a+\frac{c}{2})}{2a+c} \right\rceil \\ \hat{t}_2 = \mathrm{sgn}(r_2)\times \left\lceil \dfrac{|r_2| - \frac{b}{2}}{b} \right\rceil \end{cases} \tag{A32}$$

$$t_i = \mathrm{sgn}(\hat{t}_i)\times \min\left\{|\hat{t}_i|, 1\right\} \tag{A33}$$

$$\begin{cases} p_1 = x_1 - t_1(c+2a) \\ p_2 = x_2 - t_2 b \end{cases} \tag{A34}$$

$$\begin{cases} f_1 = (p_1+a)^2 + p_2^2 \\ f_2 = (p_1-a)^2 + p_2^2 \end{cases} \tag{A35}$$

where $x_i \in [-20,20]$.

Its true PS is

$$\begin{cases} x_1 = p_1 \\ x_2 = 0 \end{cases} \tag{A36}$$

where $x_1 \in [-a,a]$.

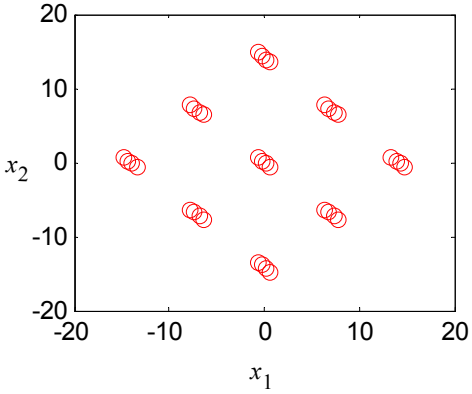
Its true PF is

$$\begin{cases} f_1 = 4a^2 v^2 \\ f_2 = 4a^2 (1-v)^2 \end{cases} \tag{A37}$$

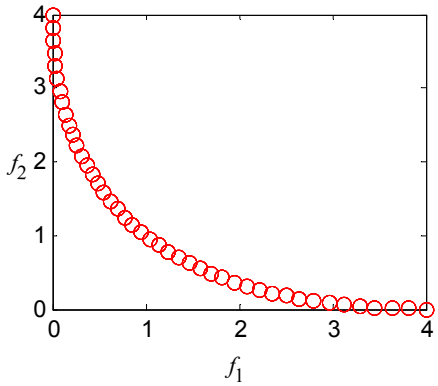
where $v \in [0,1]$.

In this paper, $w = \dfrac{\pi}{4}$, $a=1$, $b=10$, $c=8$.

Its true PS and PF are illustrated in Fig. A10.



(a) True PSs of SYM-PART rotated



(b)True PF of SYM-PART rotated

Fig. A10. Illustration of the true PSs and PF of SYM-PART rotated.

Omni-test [20]

$$\begin{cases} f_1 = \sum_{i=1}^{n} \sin(\pi x_i) \\ f_2 = \sum_{i=1}^{n} \cos(\pi x_i) \end{cases} \tag{A38}$$

where $x_i \in [0,6]$.

Its true PS is

$$x_i \in [2m+1, 2m+3/2] \tag{A39}$$

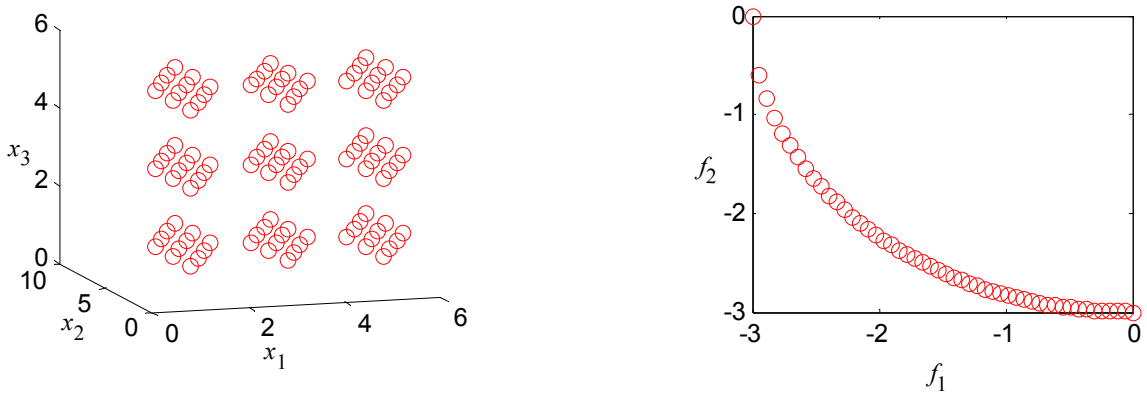
where m is integer.

Its true PF is

$$f_2 = -\sqrt{n^2 - f_1^2} \tag{A40}$$

where $-n \le f_1 \le 0$.

When $n=3$, its true PS and PF are illustrated in Fig. A11.

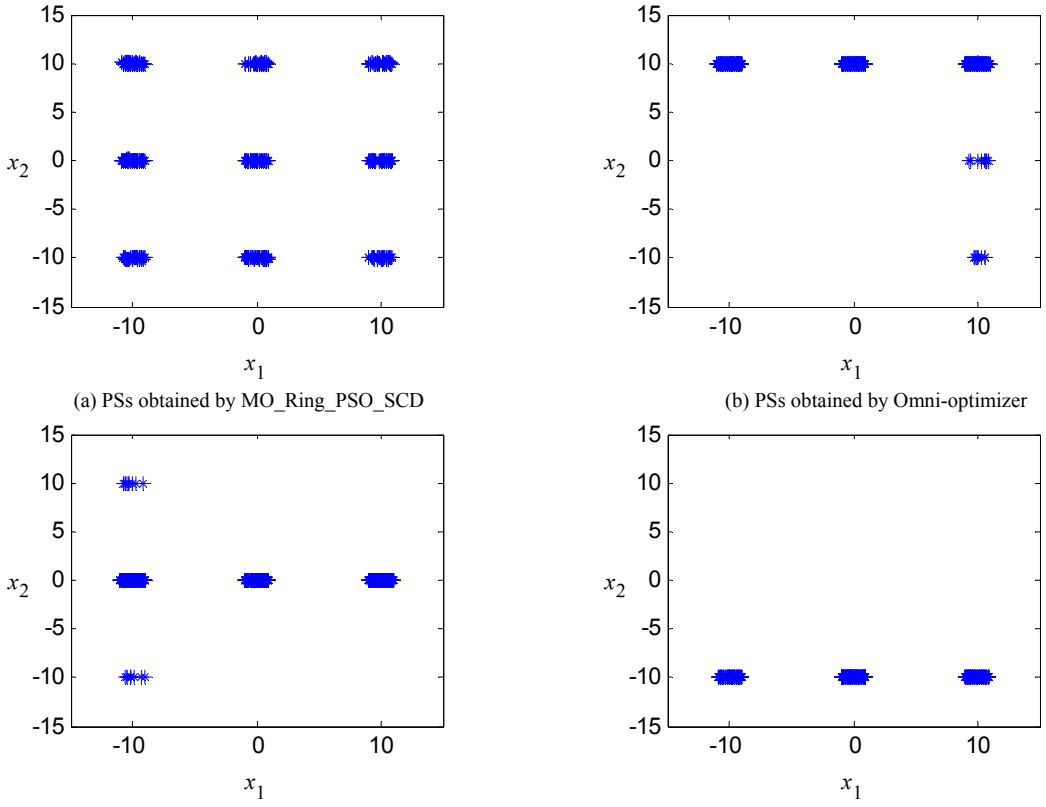


(a) True PSs of Omni-test

(b) True PF of Omni-test

Fig. A11. Illustration of the true PSs and PF of Omni-test.

B. The PSs and PF obtained by different algorithms on SYM_PART simple and Omni-test

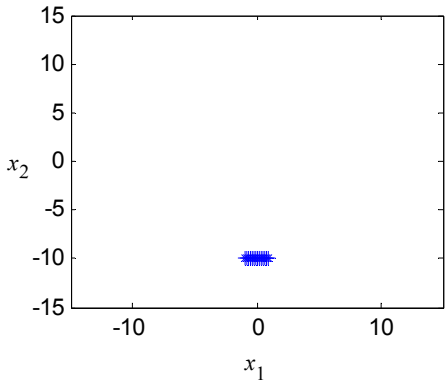


(a) PSs obtained by MO_Ring_PSO_SCD

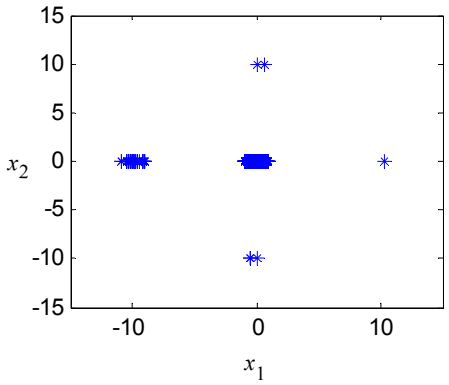
(b) PSs obtained by Omni-optimizer

(c) PSs obtained by DN-NSGAII

(d) PSs obtained by NSGAII

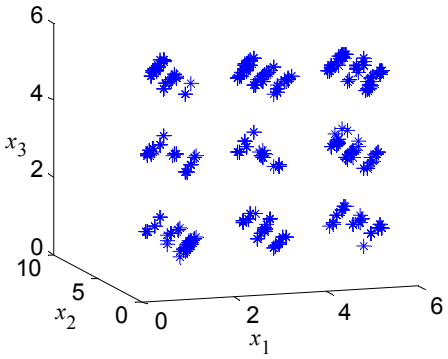


(e) PSs obtained by MOEAD

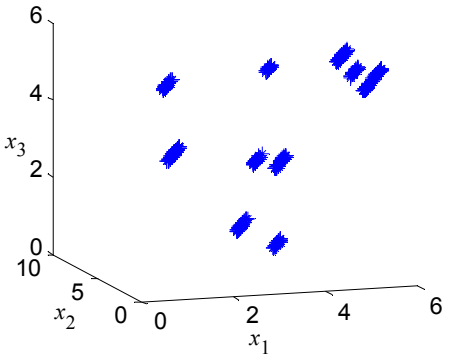


(f) PSs obtained by SPEA2

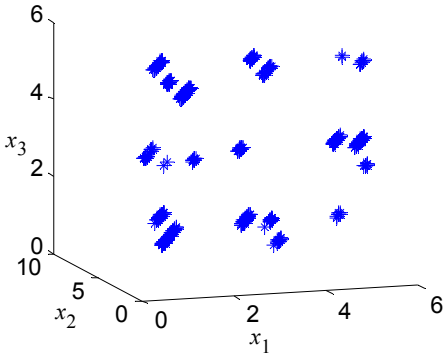
Fig. A12. The comparison of PSs obtained by different algorithms on SYM-PART simple test function.



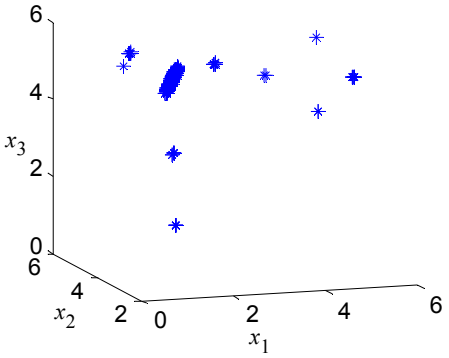
(a) PSs obtained by MO_Ring_PSO_SCD



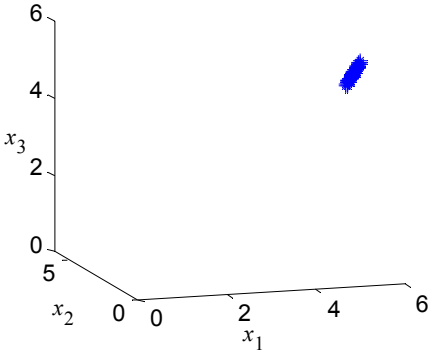
(b) PSs obtained by Omni-optimizer



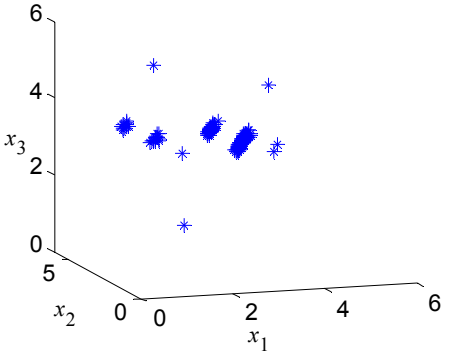
(c) PSs obtained by DN-NSGAII



(d) PSs obtained by NSGAII

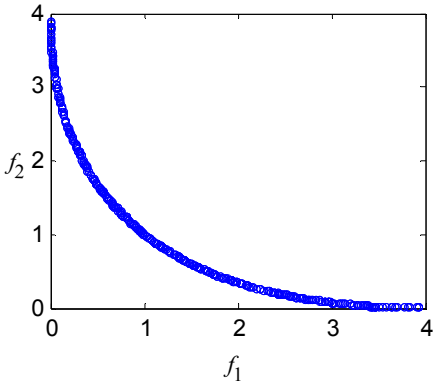


(e) PSs obtained by MOEAD

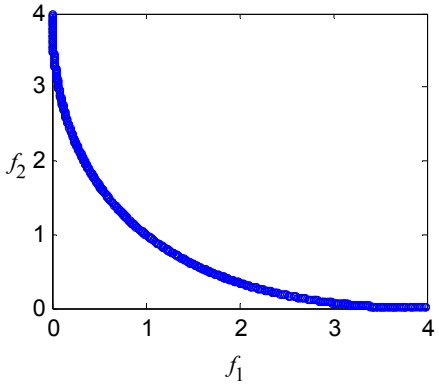


(f) PSs obtained by SPEA2

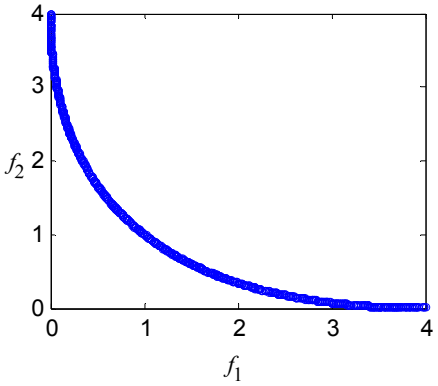
Fig. A13. The comparison of PSs obtained by different algorithms on Omni-test function.



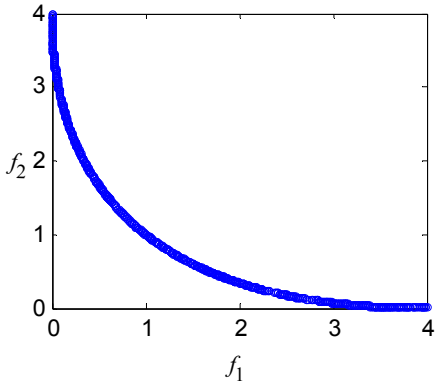
(a) PF obtained by MO_Ring_PSO_SCD



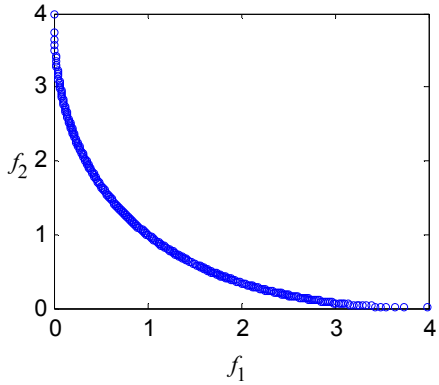
(b) PF obtained by Omni-optimizer



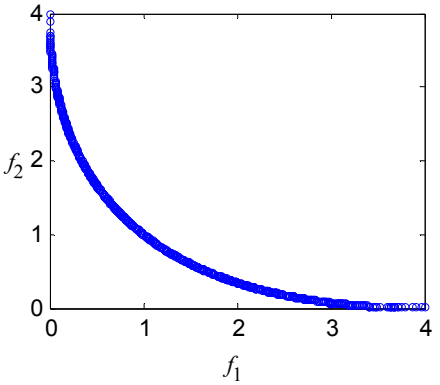
(c) PF obtained by DN-NSGAII



(d) PF obtained by NSGAII

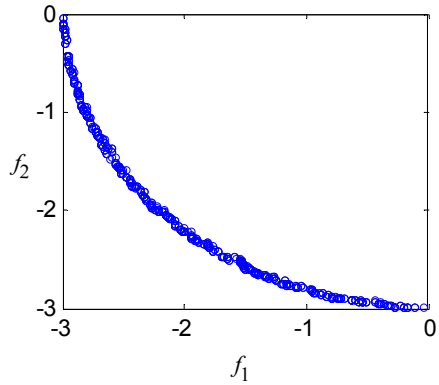


(e) PF obtained by MOEAD

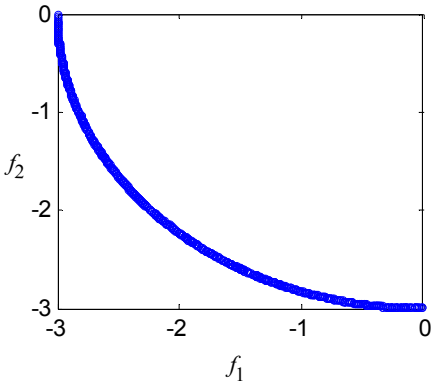


(f) PF obtained by SPEA2

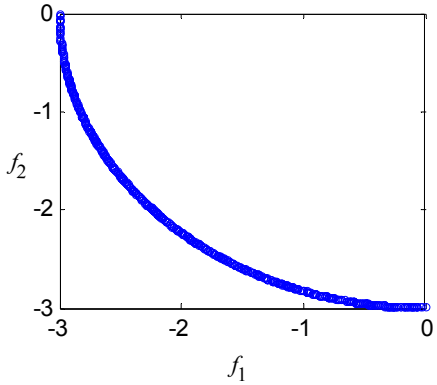
Fig. A14. The comparison of PF obtained by different algorithms on SYM-PART simple test function.



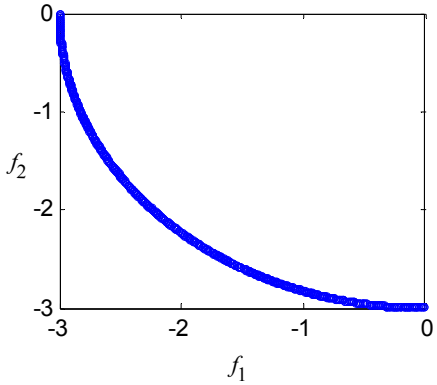
(a) PF obtained by MO_Ring_PSO_SCD



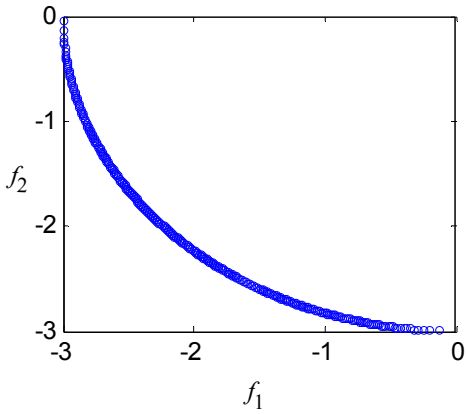
(b) PF obtained by Omni-optimizer



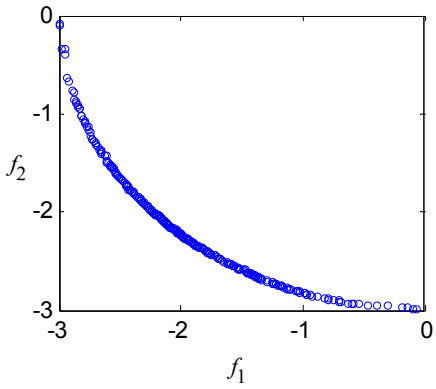
(c) PF obtained by DN-NSGAII



(d) PF obtained by NSGAII



(e) PF obtained by MOEAD



(f) PF obtained by SPEA2

Fig. A15. The comparison of PF obtained by different algorithms on Omni-test test function.