Applied Soft Computing Journal 86 (2020) 105886

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems

Boyang Qu [a], Chao Li [a], Jing Liang [b], Li Yan [a,*], Kunjie Yu [b], Yongsheng Zhu [a]

[a] School of Electronic and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China
[b] School of Electronic Engineering, Zhengzhou University, Zhengzhou 450001, China

ARTICLE INFO

Article history:
Received 27 May 2019
Received in revised form 18 September 2019
Accepted 18 October 2019
Available online 25 October 2019

Keywords:
Multimodal optimization
Multi-objective optimization
Particle swarm optimizer
Niching technique

ABSTRACT

This paper proposes a self-organized speciation based multi-objective particle swarm optimizer (SS-MOPSO) to locate multiple Pareto optimal solutions for solving multimodal multi-objective problems. In the proposed method, the speciation strategy is used to form stable niches and these niches/subpopulations are optimized to search and maintain Pareto-optimal solutions in parallel. Moreover, a self-organized mechanism is proposed to improve the efficiency of the species formulation as well as the performance of the algorithm. To maintain the diversity of the solutions in both the decision and objective spaces, SS-MOPSO is incorporated with the non-dominated sorting scheme and special crowding distance techniques. The performance of SS-MOPSO is compared with a number of the state-of-the-art multi-objective optimization algorithms on fourteen test problems. Moreover, the proposed SS-MOSPO is also employed to solve a real-life problem. The experimental results suggest that the proposed algorithm is able to solve the multimodal multi-objective problems effectively and shows superior performance by finding more and better distributed Pareto solutions.

1. Introduction

Multi-objective optimization problem (MOP) has multiple conflicting objectives which needs to be optimized simultaneously. Without loss of generality, considering that there are m objective functions and n dimensional decision variables, the minimization optimization problem can be described as:

$$\begin{aligned} \min\ & \vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x}))^T \\ s.t.\ & b_i(\vec{x}) \le 0, \qquad i = 1, 2, \dots, k \\ & e_j(\vec{x}) = 0, \qquad j = 1, 2, \dots, h \end{aligned} \tag{1}$$

Where $\vec{x}=(x_1, \dots, x_n) \in X \subset R^n$ is the n-dimensional decision vector, X is the n-dimensional decision space, $\vec{F} \in Y \subset R^m$ is the m-dimensional objective vector, Y is the m-dimensional objective space, $b_i(\vec{x})$ represents the k inequality constraints; $e_j(\vec{x}) = 0, (j = 1, 2, \dots, h)$ defines the h equality constraints. In multi-objective optimization problems, if $\vec{x}$ satisfies all of the above constraints, then $\vec{x}$ is called a feasible solution. In order to compare different feasible solutions, the Pareto dominance relationship between them is evaluated. Given two feasible solutions $\vec{x}_1$ and $\vec{x}_2$, $\vec{x}_1$ is said to dominate $\vec{x}_2$, if the relationship between $\vec{x}_1$ and $\vec{x}_2$ satisfy the formula (2):

$$\forall i = 1, 2, \dots, m,\ f_i(\vec{x}_1) \le f_i(\vec{x}_2) \wedge \exists j = 1, 2, \dots, m,\ f_i(\vec{x}_1) < f_i(\vec{x}_2) \tag{2}$$

If $\vec{x}_1$ is not dominated by any solution, $\vec{x}_1$ is called a non-dominated solution. And the set of the non-dominated solutions in the decision space is called the Pareto-optimal set (PS). The set of all the vectors in the objective space that correspond to the PS is called the Pareto front (PF).

In real-world application, many MOPs exhibit multimodal properties, i.e., there exist multiple PSs in the decision space corresponding to the same PF in the objective space. In detail the PS of the problem comprises a number of disjoint subsets in the decision space, while their corresponding PF in the objective space is the same one. And this kind of problem was defined as multimodal multi-objective problem (MMOP) by Liang et al. [1]. As shown in Fig. 1, the two elliptical regions in the decision space correspond to the same PF in the objective space, in which A and B in the decision space correspond to A* and B* in the objective space respectively. In the traditional multi-objective optimization algorithms, B will be removed because its corresponding position in the objective space B* is close to A*. However, the distance d_1 between A and B in the decision space is large and both A and B should be reserved to provide more solution options to decision maker. Hence, it is a new challenge to locate and maintain all the Pareto-optimal solutions in the decision space simultaneously.

When solving the multimodal multi-objective optimization problems, more attention should be paid to searching out the multiple Pareto-optimal solutions in the decision space. To address the multimodality of the problem, various algorithms have

* Corresponding author.
E-mail address: yanli@zut.edu.cn (L. Yan).

https://doi.org/10.1016/j.asoc.2019.105886
1568-4946/

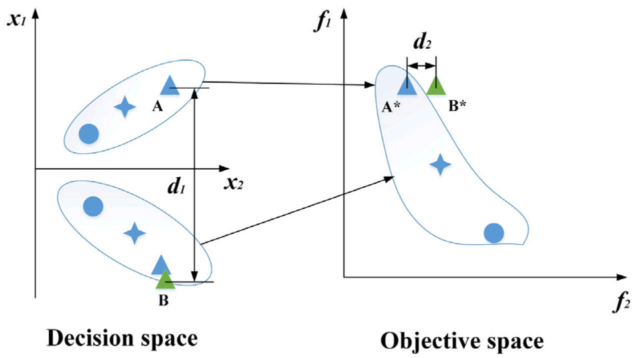


Fig. 1. Multimodal multi-objective optimization problem.

been developed [2–8] and niching techniques are commonly used in these algorithms. In the niching based algorithms [9–14], multiple stable niches are formed in the solution space to prevent the population from converging to a single solution. One simple niching method is crowding [15,16]. And this method only allows the similar individuals in the population to compete, and the similarity is generally evaluated by the Euclidean distance. Fitness sharing [17,18] is another classical niching techniques. In this method, the population is divided into different subpopulation according to the similarity of the individuals, and the individuals are forced to share information with the individuals in the same subpopulation. Although above two methods have proven to be effective in solving multimodal problems, it is difficult to set a suitable value for the niche parameters without any prior knowledge of the problem [19]. Moreover, speciation [20] is another effective niching method. The method is based on the concept of species, where the population is divided into different species according to the similarity of the individuals. And each species is grouped around a dominant individual called the species seed. This approach has the ability to form multiple stable niches to locate multiple optima simultaneously. However, the existing speciation based multimodal methods can only solve single-objective multimodal problems. To our best knowledge, a speciation based algorithm is proposed for the first time in this paper to solve MMOP.

In this paper, we propose a self-organized speciation based multi-objective particle swarm optimizer to solve the multimodal multi-objective problems. In the proposed algorithm, a niche-based speciation strategy is adopted to seek the neighborhood structure of the swarm population and to formulate different species subpopulation. And the species evolve to multiple stable niches that can locate a large number of Pareto-optimal solutions in the decision space. Meanwhile, a self-organizing mechanism is proposed to improve the efficiency of the speciation and to prevent the different species from overlapping, ensuring that more and better distributed Pareto-optimal solutions can be obtained.

The reminder of this paper is organized as follows. Section 2 reviews the related works about the multimodal multi-objective optimization. Section 3 introduces the details of the proposed SS-MOPSO. Section 4 describes the experimental settings. And Section 5 reports the experimental results and the relevant analysis. Section 6 concludes this paper.

2. Related work

2.1. Prior works on multimodal multi-objective optimization

The multi-objective optimization aims at approximating the PF in the objective space. However, the target of multimodal multi-objective optimization is not only to obtain a good PF approximation in objective space, but also to locate multiple equivalent PSs in the decision space simultaneously [21]. For addressing the multimodality in the decision space, Deb [22] proposed the Omni-optimizer to maintain the diversity in the decision space, in which the crowding distance in the decision space was considered. However, maintaining good distributions does not mean more PS can be located. Zhou [23] adopted the principal component analysis technique to estimate the PS manifold and built a probabilistic model for the distribution of the Pareto-optimal solutions.

Liang [1] proposed a decision space based niching multi-objective evolutionary algorithm (DN-NSGAII). This algorithm employs the niching method to create the mating pool, and the crowding in objective space is replaced with the crowding in decision space. The simulation results indicate that this algorithm can locate more Pareto-optimal solutions than other multi-objective optimization algorithm.

Yue [24] proposed a multi-objective particle swarm optimizer using ring topology and special crowding distance (MO_Ring_PSO_SCD), in which ring topology is adopt to form niches for locating more Pareto-optimal solutions. And a new non-dominated sorting scheme combined with the special crowding distance is proposed to balance the diversity in the both objective and decision space. However, the niche formed based on the ring topology may not represent the real neighborhood of the population individuals. Hence, the convergence accuracy of the individuals may be affected.

Liang [25] proposed a self-organizing multi-objective particle swarm optimization algorithm for solving multimodal multi-objective problems (SMPSO-MM). In SMPSO-MM, the self-organizing map network is used to establish the neighborhood of the individuals, and adopted non-dominated-sort method with special crowding distance to maintain the diversity in the decision space. The results demonstrate that this algorithm is effective for solving the multimodal multi-objective problems.

Yen [26] introduced a novel multimodal multi-objective evolutionary algorithm using two archive and recombination strategies (TriMOEA-TA&R). In TriMOEA-TA&R, the convergence and the diversity archives are adopted to guarantee the diversity in both the decision and objective space. The experimental results show that the proposed algorithm is competitive.

From the above, it can be seen that a small number of multi-objective optimization algorithms have been proposed to solve the multimodal multi-objective problems. However, how to improve the diversity in both decision space and objective space to obtain a complete and well-distributed PSs and PF is still a challenging task. Therefore, in this paper, we propose a new self-organized speciation method to increase the diversity and thus the global searching ability of the algorithm in decision space. Meanwhile, the non-dominated sorting scheme combined with a special crowding distance (SCD) are introduced to maintain the diversity of the algorithm in objective space and to select and reserve the multiple equivalent Pareto optimal solutions.

2.2. Particle swarm optimization

The particle swarm optimization (PSO) algorithm was proposed by Eberhart and Kennedy [27] in 1995, and it stems from the behavioral study of predation on birds. PSO algorithm treats the search space of the problem as a flight space and abstracts each bird into a particle to represent a candidate solution for the problem. The basic idea of the algorithm is to find the optimal solution through the cooperation and information sharing among the particles in the swarm. Compared with other evolutionary algorithms, the principle of the PSO algorithm is simple, and it has few parameters. Meanwhile, the strong memory ability makes PSO utilize the history information more sufficiently than other optimization algorithms, and the PSO has been applied on solving some complex problems, such as large scale optimization [28], many-objective optimization [29] and dynamic optimization [30]. Moreover, the most important is that PSO has been used to solve the MMOPs successfully and showed satisfactory results [31]. Hence, the PSO algorithm is a suitable choice to optimize the MMOP.

In PSO, each particle has own position vector and velocity vector, where the position of each particle is a potential solution. The personal historical best position of the particle (*pbest*) and the global best position of the particles in the swarm (*gbest*) are employed to guide the particle's flight and update its position. Moreover, PSO has global version and local version. And in the global version, the particle tracks the position of the best particles in the swarm, and the velocity update is as follows formula (3). In the local version, the particles track the position of the best particles in the topology neighborhood (*nbest*), and the velocity update is as follows formula (4). Choosing *nbest* as the leader of the particle can increase the diversity of the algorithm and thus avoids the premature convergence. So the local version of PSO is more suitable for solving the multimodal multi-objective problems. Hence, in this paper, we adopt the local version (4) to update the particles velocity. The specific update formulas are as follows:

$$v_i(t) = Wv_i(t-1) + C_1r_1(x_{pbest} - x_i(t)) + C_2r_2(x_{gbest} - x_i(t)) \quad (3)$$

$$v_i(t) = Wv_i(t-1) + C_1r_1(x_{pbest} - x_i(t)) + C_2r_2(x_{nbest} - x_i(t)) \quad (4)$$

$$x_i(t) = x_i(t-1) + v_i(t) \quad (5)$$

Where W is the inertia weight controlling the effect of the previous velocity on the current velocity. When W is larger, the former velocity has a bigger impact and the global search ability is stronger. When W is smaller, the previous velocity has less influence and the local search capability is stronger. C_1 and C_2 are referred to as learning factors. C_1 regulates the step length of the particle towards the personal best position, while C_2 controls the step length of the particle towards global best position. r_1 and r_2 are mutually independent pseudo-random numbers, subject to an uniform distribution on [0.1].

3. SS-MOPSO

The multimodal multi-objective optimization need to search out multiple Pareto-optimal solutions in the decision space. To address this issue, a Self-organized Speciation based Multi-Objective Particle Swarm Optimization algorithm is proposed. In the proposed algorithm, the concept of speciation is adopted to form multiple stable niches. And the formed multiple different species are able to evolve towards multiple optima in parallel. Meanwhile, the self-organized mechanism is proposed to control the formulation process of the species. In this section, the mechanism of the self-organized species formulation is analyzed. Then the procedure of SS-MOPSO is presented.

3.1. The proposed self-organized speciation

3.1.1. Motivation

Multiple swarms based method is frequently used to solve the multimodality of the problem [32,33]. However, when the population is divided into multiple subpopulations, it is usually demand to preset the number of individuals within the subpopulations firstly [34,35]. As shown in Fig. 2, there are nine individuals in the search space and the size of the subpopulation is preset to three. Thus the population has to be divided into three subpopulations. Pentagrams A, B and G represent the non-dominant individuals in the population. So they are chosen as the leaders for the three subpopulations respectively. Then the remaining individuals in the population are assigned to the corresponding subpopulations according to their Euclidean distances from the three non-dominant individuals. In this way, the first subpopulation is composed of the individuals A, C and D. The second subpopulation is constituted of the individuals B, E and F. And the third subpopulation is made up of the individuals H, G and I. And the three subpopulations will evolve to different PSs regions. However, the allocation of the individuals within the subpopulations may be inappropriate. Taking the individual F as an example, it should have been assigned to the third subpopulation due to the closer Euclidean distance with the non-dominant individual G. However, since the number of individuals in each subpopulation is specified, the individual F is forced to form a niche with the individual B. With the individual B as the leader, the individual F has fallen into an improper neighborhood, which may affect the convergence accuracy of the solution. Therefore, it may degrade the performance of the algorithm seriously resulting in an incomplete PSs.

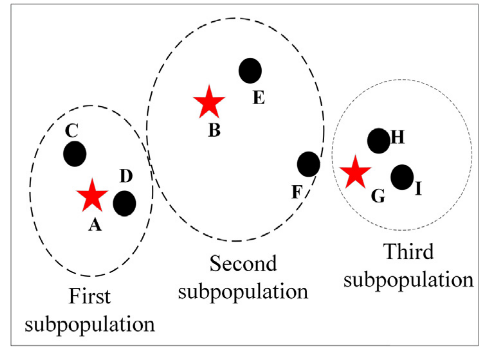


Fig. 2. Description of the subpopulation formation.

Moreover, the multiple swarms based methods have been successfully used to solve the MMOPs [36]. However, there are still some shortcomings for these methods to improve. For example, MO_Ring_PSO_SCD [24] forms multiple subpopulation by employing the ring topology structure. This method identifies the neighborhoods according to the indexes of the individuals, however, this neighborhood relationship does not represent the real distribution of the individuals in decision space, which may affect the search efficiency of the formed subpopulations. SMPSO-MM [25] forms multiple subpopulation by the self-organizing map network. Although the subpopulation formed can truly reflect the neighborhood relationship between the individuals, the self-organizing map network needs to be trained before work thus increasing the complexity of the method.

In view of the shortcomings of the above multiple swarms based methods, this paper adopts the concept of speciation to divide the population into multiple species subpopulation based on their similarity. Each species is grouped around a non-dominated individual call the species seed. And a niche radius is specified to find the similar individuals with the seed to form a species. However, in the original speciation method, all the individuals have to be calculated the distances from the seeds that have been determined to find their own species. Thus the size of the species is not determined until the last individual of the population has been assigned, which reduces the efficiency of the algorithm. In addition, during the later phase of original speciation process, there may exist many overlapping individuals between different species. And these individuals are directly assigned to the seeds with higher non-dominated ranks. Resultantly, more individuals

Algorithm 1: Pseudocode for the self-organized speciation

```
// All individuals in ascending rank values are stored into P_sorted
Generate P_sorted
NP_sorted = the number of individuals in P_sorted
while NP_sorted ≠ 0 do
//determine the species seed
species seed(j) = The first individual in P_sorted
// Speciation based on the self-organized mechanism
for i=1: NP_sorted
    if dis(species seed(j), P_sorted(i)) ≤ R      // R is the species radius
        P_sorted(i) belong to the species(j)
    end if
end for
Update P_sorted by removing the individuals in P_sorted that have been assigned to the species(j)
Update NP_sorted
end while
Output the set of species
```

are guided by the seeds with higher ranks evolving to some limited areas. The diversity of the species may be reduced making a bad influence on the distribution of the obtained PSs. Hence, this paper proposes a self-organized mechanism to help the speciation to form multiple independent species. And the details are shown in the following section.

3.1.2. The self-organized mechanism

In this paper, the self-organized mechanism is proposed. Species is formulated in a self-organized manner. The species seeds are selected according to the non-dominated ranks of the individuals. Then the seed directly determines the individuals contained in its own species according to a preset radius. Once the species seed is determined, the individuals within this species is determined too. The formation process of one species do not impact the other, and there is no overlap between them. Resultantly, the self-organized mechanism reduces the computational complexity and improves the efficiency of the species formulation. Hence, using the proposed self-organized speciation method, a more reasonable clustering results can be obtained for the population in Fig. 2. As shown in Fig. 3, the pentagram A represent the best non-dominant individual in the population, which is employed as the seed of the first species. R is the specified radius of the species. The individuals covered by R are assigned to the same species. Therefore, the first species contains the three individuals A, C and D. Among the remaining individuals, B is the best non-dominated individual, which is selected as the seed of the second species. Within the radius R, only E is included. Hence, the second species contains the two individuals B and E. Lastly, the individual G is the best individual in the remaining population. Under the influence of the proposed self-organized mechanism, the third species contains the individual G, H, I and F. Therefore, it is obvious that the species seeds are the best-fit individuals in the species and the individuals within the same species are guided by the seeds to the regions that make them even better. Therefore, with the evolution of different species, the individuals in different species are able to locate multiple PSs in the decision space.

The procedure of the self-organized speciation is shown in Algorithm 1. First, all individuals are sorted in ascending order based on their rank values obtained by the non-dominated sorting algorithm with special crowding distance [24], and stored into the $\boldsymbol{P}_{sorted}$. Then, the first individual in $\boldsymbol{P}_{sorted}$ is selected as the species seed. All individuals in the $\boldsymbol{P}_{sorted}$ are calculated the Euclidean distances with the species seed and the individuals that fall within the R distance from the species seed are classified as the same species. Then, the selected individuals are removed from $\boldsymbol{P}_{sorted}$. The above steps are repeated until the number of individuals in $\boldsymbol{P}_{sorted}$ is zero.

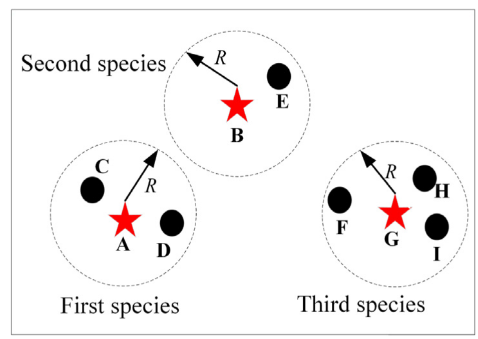


Fig. 3. Principle of the self-organized mechanism.

3.1.3. Special crowding distance

The special crowding distance (SCD) is proposed by Yue in [24] that evaluates the crowding of the particle in both the decision and objective spaces. Combined with non-dominated sorting scheme [37], SCD is employed to select and maintain the Pareto optimal solutions that correspond to the same point in the objective space. The non-dominated sorting scheme with the special crowding distance, namely Non-dominated-SCD-sort, is able to promote diversity in both spaces simultaneously. The specific procedure of the SCD involves two steps. In the first step, the crowding distance of each particle in both the decision and objective space is calculated. In the second step, the crowding distances from the first step are used to assign a SCD metric to each particle in the decision and objective spaces. More details about the calculation of SCD can be found in [24].

3.2. Procedure of SS-MOPSO

The core goal of multimodal multi-objective optimization is to locate and reserve multiple Pareto-optimal solutions in the decision space. Using the proposed self-organized speciation, the species are constructed to locate multiple different Pareto-optimal solutions in parallel. At the same time, the selection method based on Non-dominated-SCD-sort can reserve the found multiple Pareto-optimal solutions. Therefore, in theory, the proposed self-organized speciation has the ability to solve the multimodal multi-objective problems incorporated with the Non-dominated-SCD-sort.

Particles within each species update their positions through the seed of their own species. The procedure of SS-MOPSO is shown in Algorithm 2, where ***POP*** represents the entire population and $\boldsymbol{POP_i}(t)$ stands for the ith particle at the tth generation. In addition, the personal optimal archives (***POA***) is established and the personal optimal positions are stored in ***POA***, where $\boldsymbol{POA}\{i\}$ represents the ith particle's optimal positions found so far. The steps of the SS-MOPSO algorithm are shown as follows. First, the entire population ***POP*** and the personal optimal archives ***POA*** are initialized. Then, the particles are ranked by the Non-dominated-SCD-sort. After the sorting process, the self-organization speciation shown in Algorithm 1 is employed to form multiple species. Next, the particle that ranked first in $\boldsymbol{POA}\{i\}$ according to the Non-dominated-SCD-sort is chosen as the *pbest*, and the seed of its own species is chosen as the *nbest*. The $\boldsymbol{POP_i}(t)$ is updated to $\boldsymbol{POP_i}(t+1)$ according to (4) and (5). Note that if there is only one particle in a species, the species will be reserved for maintaining the diversity of the population, and this particle is selected as the seed of this species. After the evaluation, $\boldsymbol{P_i}(t+1)$ is added into $\boldsymbol{POA}\{i\}$. The particles in $\boldsymbol{POA}\{i\}$ are reordered by the Non-dominated-SCD-sort scheme. Repeat the above steps until the termination condition is reached.

Algorithm 2: SS–MOPSO

```
Initialize the population POP(0)
Evaluate (POP(0))
Fes= size of the POP
Initialize POA
for i = 1: ParticleNumber
    POA{i} = POP_i(0)
end for
while Fes < Max Fes do
    Sort all the particles in the POP by the Non-dominated-SCD-sort method
    Formulate species using the Algorithm 1
    for i = 1 : ParticleNumber
//Select pbest and nbest
      pbest_i = the first particle in sorted POA{i}
      nbest_i = the seed of its own species
      Update POP_i(t) to POP_i(t+1) according to equations (4) and (5)
      Evaluate (POP_i(t+1))
      Fes=Fes+1
//Update POA
      Put POP_i(t+1) into POA{i} and sort the solutions in POA{i}
    end for
end while
Output the non-dominated particles in POA
```

3.3. Complexity

The complexity of the proposed algorithm for forming species (Algorithm 1) can be calculated based on the number of evaluations of Euclidean distances between the two individuals. Assuming that there are N individuals sorted and stored in $\boldsymbol{P_{sorted}}$ whose size is *NP_sorted*, the **for** loop goes through *NP_sorted* times to see if the individual is within the radius R of the current seeds. The best case is that all the individuals are within the radius R of the first seed, and thus the **for** loop is only executed N times. At worst the **for** loop is executed $N(N+1)/2$ times when all the N individuals are the seeds of N independent species. Hence, the number of Euclidean distance calculations required for this procedure, $T(N)$, can be obtained by the following:

$$N \leq T(N) \leq \frac{N(N+1)}{2} \tag{6}$$

which shows that the highest complexity of the procedure is $O(N^2)$. The speciation based multi-objective particle swarm optimizer (S-MOPSO) has the same maximum and minimum complexity as the proposed SS-MOPSO [38]. However, it is obviously that the two extreme cases are hard to appear. And in the common case, the complexities of S-MOPSO and SS-MOPSO are different, and the proposed SS-MOPSO is more efficient than the original S-MOPSO. For S-MOPSO, the complexity for forming multiple species is related not only to N, but also to the number of seeds N_s. The individuals are required to compute the Euclidean distances from all the seeds that have been found. Thus, the number of evaluations of Euclidean distances required by S-MOPSO increases continuously with the process of the species formulation. So the complexity of S-MOPSO can be expressed by $O(N_s \cdot N)$ in the common case. However, in the proposed SS-MOPSO, the individuals that have been assigned to some species is removed from the population, and the remaining individuals of the population no longer calculate the Euclidean distances from the previous seeds. Hence, its complexity only depends on N, and the number of Euclidean distance calculations required by SS-MOPSO is decreases rapidly in the process of speciation. Thus the complexity of SS-MOPSO is lower than S-MOPSO in the common case.

4. Experimental settings

4.1. Test functions

In this paper, we used fourteen test functions to test the performance of SS-MOPSO including eight MMF problems [24], three SYM-PART problems [39] and three Omni-test problems (with different dimensions) [22]. In the eight MMF problems, the numbers of PSs are two for MMF1, MMF2, MMF3 and MMF7 and the rest are four. In the SYM-PART problems, the numbers of PSs are all nine. In addition, the Omni-test problems with the dimensions 3, 4 and 5 were used, and the numbers of PSs are 27, 72 and 360 for them respectively. As a result, the three Omni-test problems are the toughest problems among all the fourteen test functions. Note that the dimensions of all the other problems are two.

4.2. Performance indicators

We used the inverted generational distance *IGDX* [23], Hypervolume (*Hv*) [40] and the Pareto Sets Proximity (*PSP*) [24] for assessing the performance of SS-MOPSO. The *IGDX* represents the average distance (Euclidean distance) between the obtained solutions and reference solutions (true PS) in the decision space. Let S^* denote a set of uniformly distributed reference points along the true PS and K denote a set of non-dominated solutions of the final population. The *IGDX* can be computed as the average distance from S^* to K in the decision space as follows :

$$IGDX(K, S^*) = \frac{\sum_{c \in S^*} \min\_d(c, K)}{|S^*|} \tag{7}$$

Table 1
PSP values obtained by different algorithms.

	SS-MOPSO mean ± std dev	S-MOPSO mean ± std dev	SMPSO-MM mean ± std dev	MRPS mean ± std dev	DN-NSGAII mean ± std dev	MOEAD mean ± std dev	MO_PSO mean ± std dev
MMF1	84.85 ± 2.75	83.11 ± 3.28(+)	**86.05** ± **3.57**(=)	67.00 ± 2.74(+)	42.68 ± 5.37(+)	7.17 ± 3.18(6+)	2.99 ± 0.48(+)
MMF2	**163.30** ± **18.06**	103.33 ± 24.22(+)	149.81 ± 17.34(+)	104.11 ± 18.89(+)	61.52 ± 33.32(+)	4.26 ± 3.19(+)	4.78 ± 2.14(+)
MMF3	**189.94** ± **12.67**	147.91 ± 21.20(+)	187.97 ± 17.18(=)	141.19 ± 17.84(+)	75.11 ± 28.83(+)	7.00 ± 3.89(+)	5.66 ± 1.77(+)
MMF4	**139.40** ± **3.49**	135.40 ± 5.20(+)	138.86 ± 3.83(=)	114.51 ± 4.40(+)	38.42 ± 8.77(+)	1.84 ± 1.02(+)	3.08 ± 0.60(+)
MMF5	**40.34** ± **1.43**	40.11 ± 1.21(=)	40.07 ± 1.38(=)	33.25 ± 1.27(+)	14.57 ± 1.49(+)	3.37 ± 1.46(+)	2.14 ± 0.38(+)
MMF6	41.92 ± 1.07	**42.40** ± **1.38**(=)	41.72 ± 1.22(=)	36.54 ± 1.11(+)	18.24 ± 1.73(+)	4.98 ± 2.39(+)	2.46 ± 0.35(+)
MMF7	123.73 ± 6.80	133.36 ± 6.23(−)	**142.37** ± **4.57**(−)	107.72 ± 4.49(+)	95.78 ± 17.17(+)	7.53 ± 4.80(+)	3.73 ± 1.02(+)
MMF8	**62.93** ± **4.45**	59.34 ± 2.42(+)	62.37 ± 2.82(=)	47.31 ± 1.97(+)	18.00 ± 5.90(+)	0.13 ± 0.13(+)	1.42 ± 0.37(+)
SYM-PART1	**34.50** ± **1.62**	13.34 ± 1.03(+)	31.25 ± 3.31(+)	21.81 ± 1.54(+)	0.45 ± 0.59(+)	0.01 ± 0.02(+)	0.51 ± 0.39(+)
SYM-PART2	**30.77** ± **1.98**	12.59 ± 1.29(+)	16.91 ± 1.34(+)	18.14 ± 1.22(+)	3.06 ± 4.01(+)	0.01 ± 0.01(+)	0.63 ± 0.58(+)
SYM-PART3	**28.17** ± **2.97**	8.04 ± 3.45(+)	15.78 ± 2.80(+)	9.80 ± 4.70(+)	5.57 ± 9.97(+)	0.01 ± 0.01(+)	0.46+0.34(+)
Omni-test1	**14.27** ± **1.53**	6.75 ± 1.10(+)	7.74 ± 1.55(+)	7.70 ± 1.59(+)	1.16 ± 0.23(+)	0.03 ± 0.01(+)	0.65 ± 0.13(+)
Omni-test2	**1.46** ± **0.13**	0.84 ± 0.05(+)	0.95 ± 0.07(+)	0.97 ± 0.07(+)	0.44 ± 0.08(+)	0.03 ± 0.01(+)	0.40+0.07(+)
Omni-test3	**0.58** ± **0.02**	0.47 ± 0.02(+)	0.49 ± 0.02(+)	0.50 ± 0.02(+)	0.28 ± 0.09(+)	0.02 ± 0.01(+)	0.30+0.05(+)
Sumup of All	+\=\−	11\2\1	7\6\1	14\0\0	14\0\0	14\0\0	14\0\0

Table 2
IGDX values obtained by different algorithms.

	SS-MOPSO mean ± std dev	S-MOPSO mean ± std dev	SMPSO-MM mean ± std dev	MRPS mean ± std dev	DN-NSGAII mean ± std dev	MOEAD mean ± std dev	MO_PSO mean ± std dev
MMF1	0.0118 ± 3.74e−4	0.0120 ± 4.91e−4(+)	**0.0116** ± **4.79e−4**(=)	0.0149 ± 6.09e−4(+)	0.0237 ± 3.30e−3(+)	0.1504 ± 6.68e−2(+)	0.2938 ± 3.62e−2(+)
MMF2	**0.0061** ± **6.72e−4**	0.0101 ± 2.68e−3(+)	0.0067 ± 1.33e−3(+)	0.0098 ± 0.02e−1(+)	0.0218 ± 1.50e−2(+)	0.2653 ± 1.16e−1(+)	0.1846 ± 6.44e−2(+)
MMF3	**0.0052** ± **3.45e−4**	0.0068 ± 1.03e−3(+)	0.0053 ± 4.66e−4(=)	0.0071 ± 8.93e−4(+)	0.0156 ± 7.10e−3(+)	0.1391 ± 4.66e−2(+)	0.1448 ± 3.29e−2(+)
MMF4	**0.0072** ± **1.80e−4**	0.0074 ± 2.88e−4(+)	0.0072 ± 1.98e−4(=)	0.0087 ± 3.39e−4(+)	0.0274 ± 6.30e−3(+)	0.4262 ± 1.25e−1(+)	0.2764 ± 4.09e−2(+)
MMF5	0.0248 ± 8.73e−4	**0.0247** ± **7.31e−4**(=)	0.0250 ± 8.57e−4(=)	0.0301 ± 0.01e−1(+)	0.0691 ± 7.10e−3(+)	0.2989 ± 1.26e−1(+)	0.4079 ± 5.20e−2(+)
MMF6	0.0238 ± 6.00e−4	**0.0236** ± **7.60e−4**(=)	0.0240 ± 7.05e−4(=)	0.0273 ± 8.23e−4(+)	0.0552 ± 5.60e−3(+)	0.2048 ± 7.08e−2(+)	0.3480 ± 3.28e−2(+)
MMF7	0.0081 ± 4.18e−4	0.0075 ± 3.50e−4(−)	**0.0070** ± **2.13e−4**(−)	0.0093 ± 3.82e−4(+)	0.0108 ± 2.40e−3(+)	0.1514 ± 6.62e−2(+)	0.1786 ± 2.57e−2(+)
MMF8	**0.0159** ± **1.20e−3**	0.0168 ± 6.83e−4(+)	0.0160 ± 7.19e−4(=)	0.0211 ± 8.81e−4(+)	0.0646 ± 3.52e−2(+)	2.7119 ± 5.86e−1(+)	0.6471 ± 1.76e−1(+)
SYM-PART 1	**0.0290** ± **1.38e−3**	0.0753 ± 6.06e−3(+)	0.0323 ± 3.46e−3(+)	0.0460 ± 3.30e−3(+)	3.0252 ± 1.07(+)	12.6890 ± 2.72(+)	2.4684 ± 9.75e−1(+)
SYM-PART 2	**0.0326** ± **2.13e−3**	0.0799 ± 1.05e−2(+)	0.0590 ± 4.58e−3(+)	0.0552 ± 0.04e−1(+)	1.0511 ± 0.82(+)	13.1139 ± 2.56(+)	2.1363 ± 1.08(+)
SYM-PART 3	**0.0357** ± **3.93e−3**	0.2242 ± 2.68e−1(+)	0.0651 ± 1.33e−2(+)	0.2057 ± 2.66e−1(+)	0.9556 ± 7.56e−1(+)	10.3697 ± 2.49(+)	2.2330 ± 0.95(+)
Omni-test1	**0.0710** ± **1.09e−2**	0.1514 ± 2.74e−2(+)	0.1346 ± 3.16e−2(+)	0.1365 ± 0.38e−1(+)	0.8896 ± 1.65e−1(+)	3.4183 ± 4.13e−1(+)	1.4509 ± 2.40e−1(+)
Omni-test2	**0.6882** ± **5.78e−2**	1.1725 ± 6.89e−2(+)	1.0473 ± 7.62e−2(+)	1.0253 ± 0.73e−1(+)	2.1976 ± 2.92e−1(+)	3.9763 ± 4.83e−1(+)	2.2401 ± 2.17-1(+)
Omni-test3	**1.7222** ± **6.37e−2**	2.1100 ± 8.49e−2(+)	2.0288 ± 8.15e−2(+)	1.9681 ± 0.86e−1(+)	3.1630 ± 5.12e−1(+)	4.6964 ± 4.55e−1(+)	2.9394 ± 2.52e−1(+)
Sumup of All	+\=\−	11\2\1	7\6\1	14\0\0	14\0\0	14\0\0	14\0\0

where c represents a reference point in S^*, min_$d(c, K)$ is the minimum Euclidean distance between c and the points in K. The *IGDX* values reflect the diversity and convergence of obtained solutions in the decision space. The smaller the value, the better the performance of the algorithm. While *IGDX* measures the quality of solutions in the decision space, Hypervolume considers the volume of the objective space dominated by the obtained solutions and evaluates the universality and uniformity of obtained solutions. Larger Hypervolume value are desirable.

$$Hv(P, r) = VOL(\bigcup_{x \in P} [f_1(x), r_1] \times \cdots [f_m(x), r_m]) \tag{8}$$

Where $\boldsymbol{r} = (r_1, \ldots, r_\mathrm{m})$ is the reference point in the objective space dominated by any Pareto front, and $\boldsymbol{VOL}(.)$ is the Lebesgue measure. The *PSP* indicator was first proposed by Yue [24] to reflect the similarity between the obtained PSs and the true PSs. The *PSP* value be calculated as follows:

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

where *CR* shows overlap ratio between true Pareto-optimal solutions and obtained Pareto-optimal solutions, and *IGDX* is the inverted generational distance in the decision space. Larger *PSP* value is satisfactory.

4.3. Comparison with other algorithms

To demonstrate the effectiveness of SS-MOPSO, the SS-MOPSO is compared with the following five state-of-the-art algorithms, MO_PSO [41], MOEAD [42], MO_Ring_PSO_SCD [24], DN-NSGAII [1], and SMPSO-MM [25]. MO_PSO is the original multi-objective PSO algorithm, and MOEAD is a popular multi-objective algorithm based on decomposition. MRPS (acronym of MO_Ring_PSO_SCD), SMPSO-MM and DN-NSGAII are the recently proposed multi-modal multi-objective optimization algorithm. In addition, the original speciation method [38] was also integrated with MOPSO (S-MOPSO) to compare with the SS-MOPSO in order to verify the performance of the proposed self-organizing mechanism. It should be noted that all the components of S-MOPSO is the same with the SS-MOPSO expect for the species formulation method.

For the purpose of fair comparison, the maximal number of evaluations is set to 80 000 and the population size is set to 800 for all the algorithms. Each algorithm is run 50 times on each test instance. In SS-MOPSO, both C_1 and C_2 are set to 2.05 [43] and W is set to 0.7298 [43]. Generally, the species radius is set to a value between 1/20 to 1/10 of the allowed variable range [38]. In this paper, 1/20 of the allowed variable range is employed as the species radius. And the effect of different species radius on the algorithm performance is also verified and analyzed in Section 5.4.

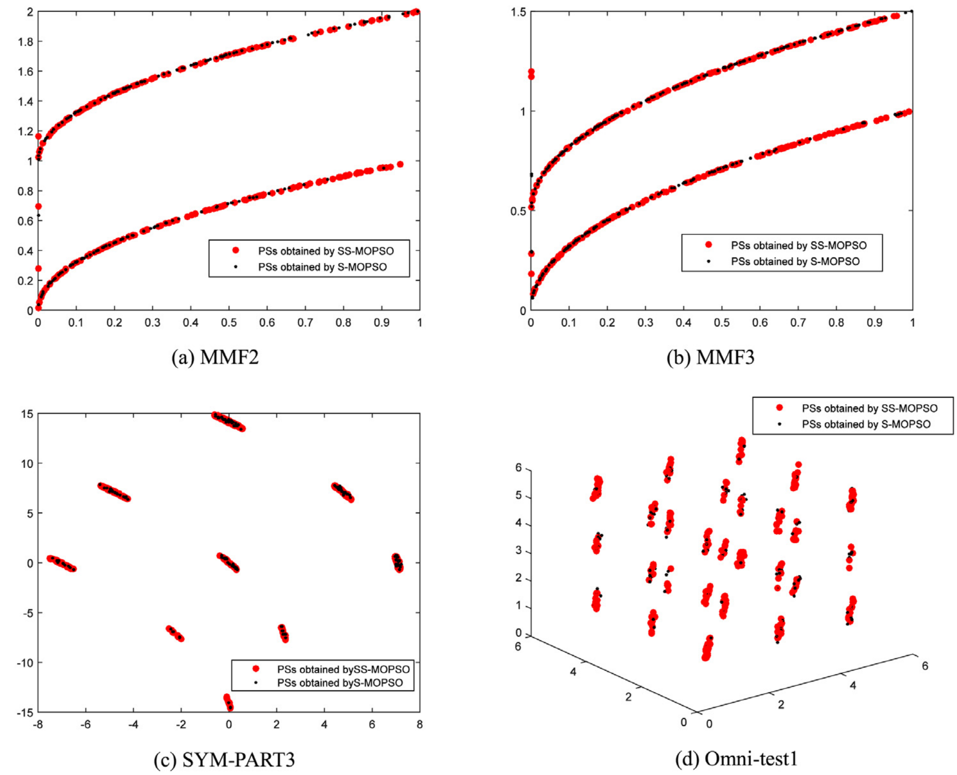


Fig. 4. The comparison of PSs obtained by SS-MOPSO and S-MOPSO.

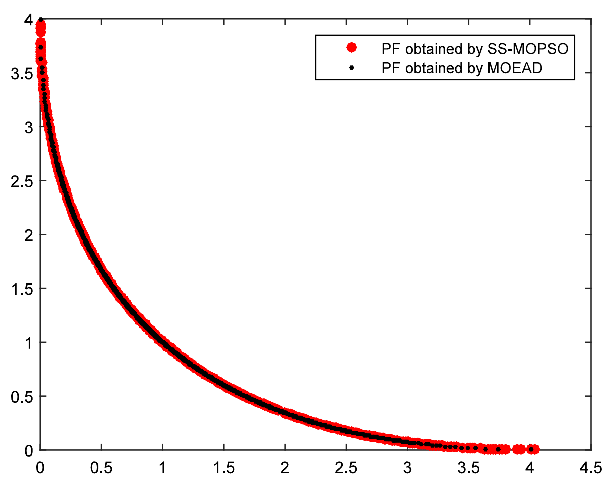


Fig. 5. Comparison of the Pareto fronts obtained by different methods on SYM-PART1.

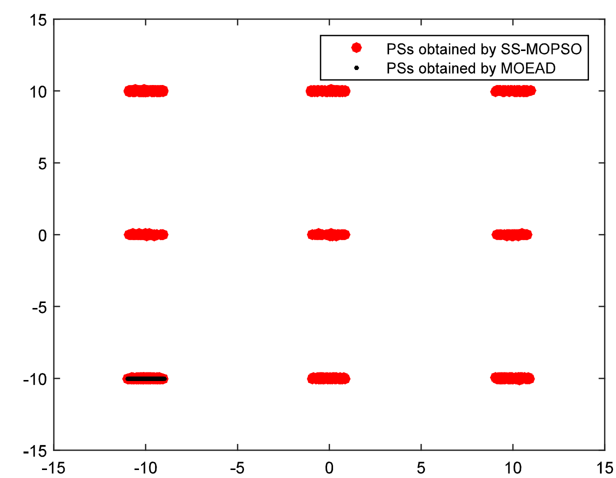


Fig. 6. Comparison of the Pareto optimal sets obtained by different methods on SYM-PART1.

5. Experimental results and analysis

5.1. The results of PSP value on algorithms

The mean *PSP* values for all the algorithms on fourteen test functions are presented in Table 1. In addition, the Wilcoxon's rank sum test is employed to indicate significance between the results obtained by SS-MOPSO and other compared algorithms at the significant level of 0.05. The hypothesis test results are shown in the parentheses. '+' ('—') indicates that SS-MOPSO shows significantly better (worse) performance in the comparison. '=' indicates that there is no significant difference between the compared results.

As shown in Table 1, the sumup results of the hypothesis test indicates that SS-MOPSO achieves the best *PSP* performance among all the compared algorithms. And Table 1 shows that SS-MOPSO and S-MOPSO behave similarly in the decision space on MMF5 and MMF6. As for MMF7, SS-MOPSO performs worse than S-MOPSO and SMPSO-MM since the distribution of the MMF7 solution is irregular. In addition, the *PSP* values obtained by SS-MOPSO performs significantly better than that of S-MOPSO on the remaining test functions. Meanwhile, the PSs obtained by SS-MOPSO and S-MOPSO on some representative problems MMF2, MMF3, SYM-PART3 and Omni-test1 are shown in Fig. 4(a)–(d). And it can be seen that the PSs obtained by SS-MOPSO are more complete and more uniform than that of S-MOPSO. The results verify the analysis in Section 3.1.1. For the S-MOPSO, the

Table 3
Hv values obtained by different algorithms.

	SS-MOPSO mean ± std dev	S-MOPSO mean ± std dev	SMPSO-MM mean ± std dev	MRPS mean ± std dev	DN-NSGAII mean ± std dev	MOEAD mean ± std dev	MO_PSO mean ± std dev
MMF1	3.66 ± 4.55e−4	3.66 ± 4.42e−4	**3.67 ± 3.99e−4**	3.66 ± 4.63e−4	3.66 ± +1.30e−3	3.67 ± 4.79e−4	3.53 ± 5.21e−2
MMF2	3.66 ± 4.20e−3	3.65 ± 8.48e−3	3.66 ± 4.16e−3	3.65 ± 8.10e−3	3.66 ± 6.20e−3	**3.67 ± 5.35e−4**	3.40 ± 1.11e−1
MMF3	3.66 ± 3.28e−3	3.66 ± 3.54e−3	3.66 ± 3.34e−3	3.65 ± 5.10e−3	3.66 ± 4.90e−3	**3.67 ± 7.22e−4**	3.45 ± 1.03e−1
MMF4	3.33 ± 8.45e−4	3.33 ± 8.91e−4	3.33 ± 8.88e−4	3.33 ± 1.20e−3	3.33 ± 3.60e−4	**3.33 ± 1.63e−4**	3.03 ± 1.18e−1
MMF5	**3.67 ± 3.03e−4**	3.66 ± 4.44e−4	3.67 ± 3.83e−4	3.66 ± 3.52e−4	3.67 ± 2.70e−3	3.67 ± 6.81e−4	3.53 ± 8.03e−2
MMF6	3.66 ± 3.19e−4	3.66 ± 3.74e−4	**3.67 ± 3.41e−4**	3.66 ± 3.31e−4	3.66 ± 1.10e−3	3.67 ± 5.53e−4	3.54 ± 5.66e−2
MMF7	3.66 ± 2.88e−4	3.66 ± 3.00e−4	3.67 ± 1.99e−4	3.67 ± 2.38e−4	3.66 ± 7.10e−4	3.67 ± 2.39e−4	3.52 ± 4.30e−2
MMF8	3.21 ± 7.30e−4	3.21 ± 1.58e−3	3.21 ± 1.15e−3	3.21 ± 1.50e−3	3.21 ± 1.30e−3	**3.21 ± 1.44e−4**	2.95 ± 1.44e−1
SYM-PART 1	1.32 ± 8.30e−4	1.27 ± 4.86e−3	1.31 ± 1.47e−3	1.30 ± 1.90e−3	1.32 ± 3.09e−4	**1.32 ± 8.35e−5**	0.85 ± 2.27e−1
SYM-PART 2	1.31 ± 1.12e−3	1.27 ± 4.39e−3	1.29 ± 2.90e−3	1.29 ± 2.20e−3	**1.32 ± 4.92e−4**	1.32 ± 6.85e−4	0.75 ± 2.67e−1
SYM-PART 3	1.31 ± 2.04e−3	1.25 ± 6.48e−3	1.28 ± 3.93e−3	1.29 ± 2.50e−3	**1.32 ± 5.00e−4**	1.32 ± 1.40e−3	0.78 ± 2.97e−1
Omni-test1	62.02 ± 5.33e−3	61.95 ± 1.56e−2	61.98 ± 1.18e−2	61.97 ± 1.24e−2	**62.06 ± 3.98e−4**	62.06 ± 4.80e−3	60.90 ± 7.48e−1
Omni-test2	77.35 ± 2.53e−2	76.84 ± 1.09e−1	77.01 ± 6.65e−2	77.03 ± 8.52e−2	**77.54 ± 2.10e−3**	77.53 ± 9.80e−3	75.47 ± 8.12e−1
Omni-test3	93.86 ± 1.08e−1	92.29 ± 3.06e−1	92.67 ± 2.65e−1	92.88 ± 2.72e−1	**94.59 ± 8.80e−3**	94.55 ± 2.42e−2	90.68 ± 9.06e−1

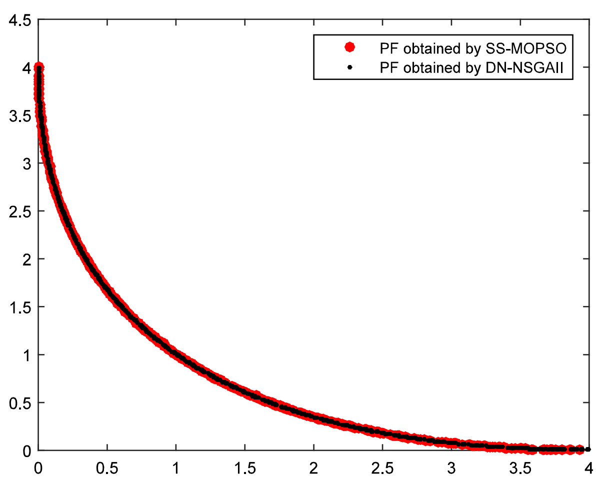


Fig. 7. Comparison of the Pareto fronts obtained by different methods on SYM-PART2.

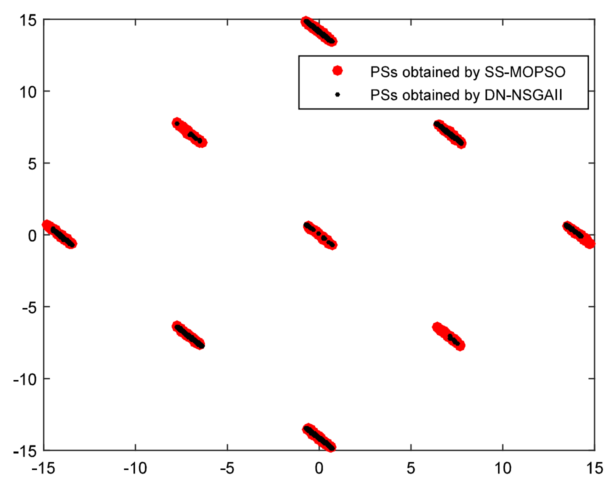


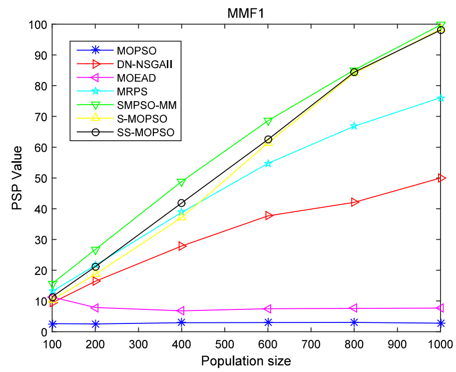
Fig. 8. Comparison of the Pareto optimal sets obtained by different methods on SYM-PART2.

allocation of the overlapping particles between different species depend on the non-dominant ranks of the seeds leading that more particles are preferred to the seeds with higher ranks. That may affect the diversity of the species. Thus, the PSs obtained by S-MOPSO is incomplete and nonuniform.

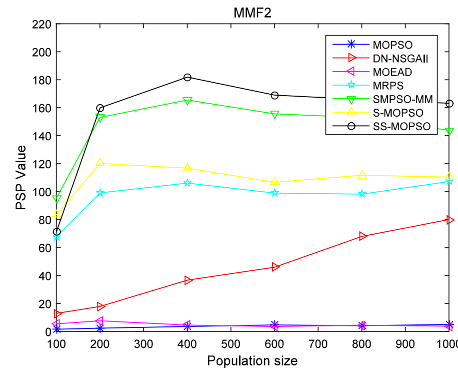
Compared with the other reported multimodal multi-objective or multi-objective algorithms, as shown in Table 1, the proposed SS-MOPSO outperforms MRPS, DN-NSGAII, MOEAD and MO_PSO greatly on all the considered test functions. SMPSO-MM achieves the best *PSP* values on MMF1 and MMF7. With respect to MMF1, the *PSP* value obtained by SMPSO-MM is a bit larger than SS-MOPSO, however, there is no significant difference between them. As for MMF7, the SS-MOPSO shows worse performance than SMPSO-MM, the reason may be that the SMPSO-MM adopted the SOM network to establish the neighborhood of the particles, which makes the SMPSO-MM more suitable for solving complex problem. In addition, SS-MOPSO shows significantly better performance than SMPSO-MM on MMF2, SYM-PART(1)–(3) and Omni-test(1)–(3). Thus, SS-MOPSO is capable of locating more equivalent Pareto-optimal solutions and obtaining a better distribution in the decision space compared with other algorithms. The reason is that the proposed SS-MOPSO employs the speciation strategy to form stable niches in the decision space. Individuals within the same species can be made to follow the species seed and be attracted to positions that make them even fitter. Hence, the SS-MOPSO can locate more Pareto-optimal solutions. Moreover, the self-organized mechanism is introduced to enhance the global search ability of the algorithm, ensuring that better distributed Pareto-optimal solutions can be obtained.

5.2. The results of IGDX and Hv values on algorithms

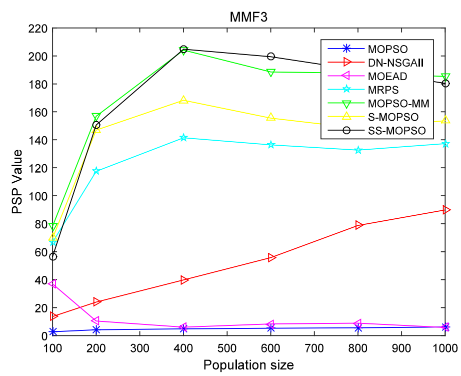
The *IGDX* values of all the algorithms on fourteen test functions are shown in Table 2. Meanwhile, the hypothesis test results between SS-MOPSO and other algorithms are also shown. As can be seen from Table 2, the *IGDX* values of SS-MOPSO are best on most of the test problems. And SS-MOPSO achieves the best *IGDX* performance among all the compared algorithms in terms of the sumup results of hypothesis test. SMPSO-MM and S-MOPSO achieve significantly better performance than SS-MOPSO on MMF7. S-MOPSO obtains the highest *IGDX* values on MMF5 and MMF6, and SMPSO-MM obtains the best *IGDX* values on MMF1. However, it can be seen from the hypothesis test results that the *IGDX* values of SS-MOPSO on the MMF1, MMF5 and MMF6 are no significant difference with SMPSO-MM and S-MOPSO. Moreover, SS-MOPSO outperforms MRPS, DN-NSGAII, MOEAD and MO_PSO significantly on all the fourteen test functions.



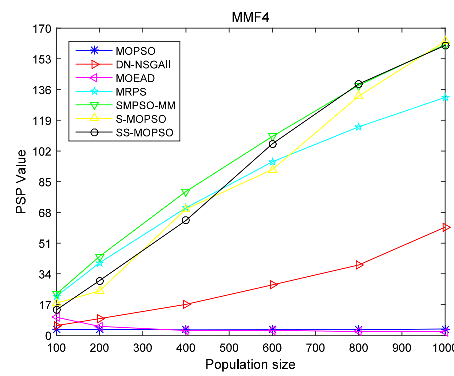
(a) *PSP* values on MMF1 with different population sizes



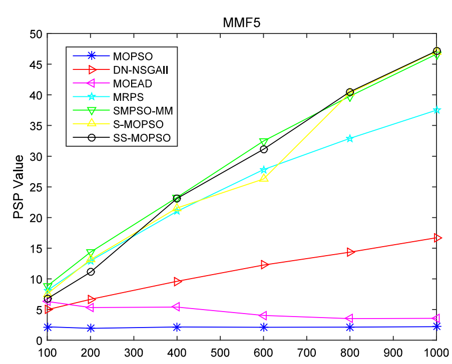
(b) *PSP* values on MMF2 with different population sizes



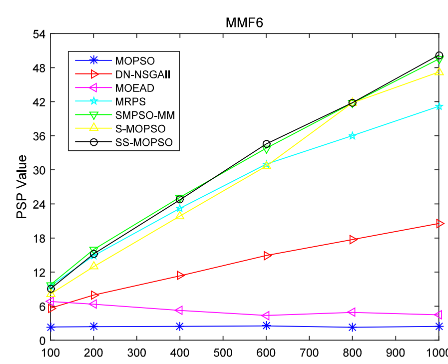
(c) *PSP* values on MMF3 with different population sizes



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



(e) *PSP* values on MMF5 with different population sizes

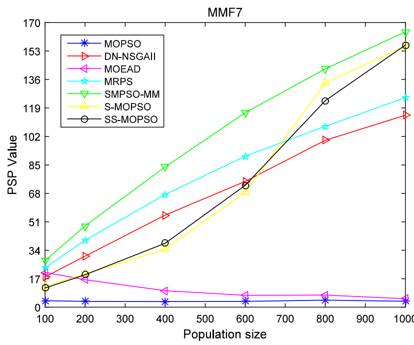


(f) *PSP* values on MMF6 with different population sizes

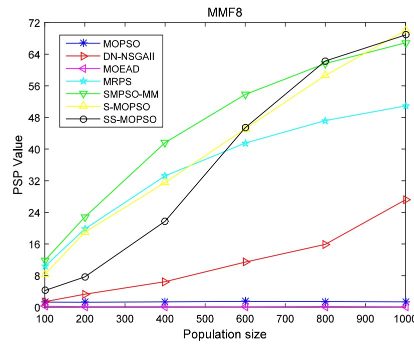
Fig. 9. PSP values with different population sizes.

Table 3 shows that SS-MOPSO has the best *Hv* value on MMF5. The *Hv* values of SMPSO-MM on MMF1, MMF6 and MMF7 are the highest among all the algorithms. DN-NSGAII obtains the highest *Hv* values on SYM-PART2, SYM-PART3 and the three Omni-test problems. MOEAD achieves the best *Hv* values on MMF2, MMF3, MMF4, MMF8 and SYM-PART1. Although SS-MOPSO does not show the best performance compared to the other algorithms, they are approximate to each other from the statistical average results. The reason is that all the algorithms are multi-objective algorithms that take into account the distribution of the solutions in the objective space.

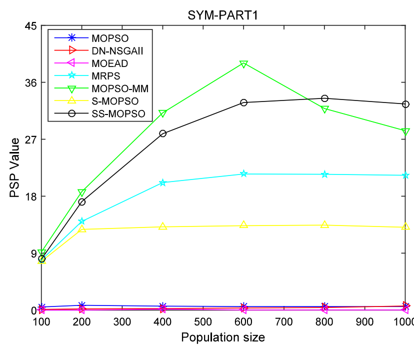
In order to demonstrate the performance of the proposed SS-MOPSO in both the decision and objective spaces, the PSs and PFs of SYM-PART1 and SYM-PART2 are taken as the examples for analysis. Fig. 5 shows the distribution of the PF obtained by SS-MOPSO and MOEAD on SYM-PART1. There are no significant differences between the results of SS-MOPSO and MOEAD. Both algorithms can get well distributed Pareto fronts. Fig. 6 shows the distribution of the Pareto-optimal solutions obtained by SS-MOPSO and MOEAD on SYM-PART1. We can clearly see that the SS-MOPSO obtains nine Pareto-optimal subsets, while MOEAD obtains only one. Figs. 7 and 8 show the distribution of the PFs and the PSs obtained by SS-MOPSO and DN-NSGAII on SYM-PART2 respectively. It can be seen that both the two algorithms obtain a well-distributed PF on the SYM-PART2. And compared with the DN-NSGAII, the Pareto-optimal solutions obtained by SS-MOPSO is more complete and more uniform. Hence, the results demonstrate that SS-MOPSO is competitive with other six algorithms for solving multimodal multi-objective problems, even though it is not the best performer in the objective space.



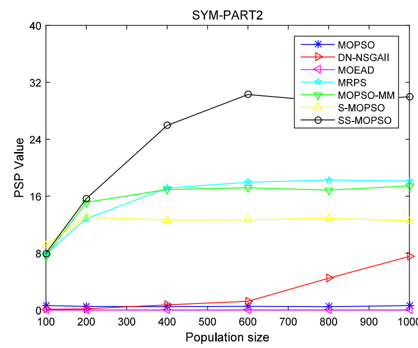
(g) *PSP* values on MMF7 with different population sizes



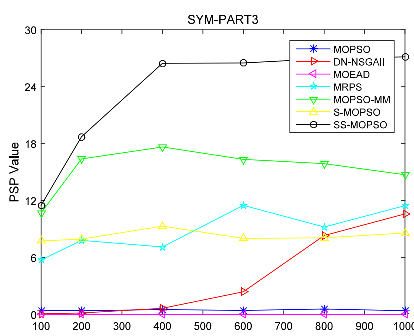
(h) *PSP* values on MMF8 with different population sizes



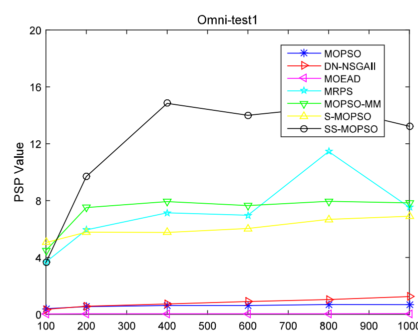
(i) *PSP* values on SYM-PART1 with different population sizes



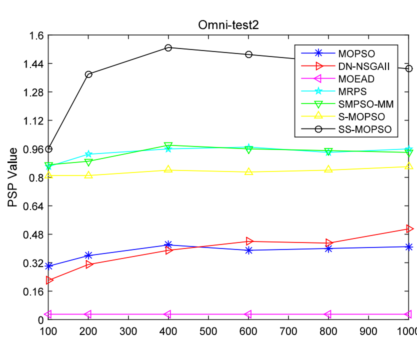
(j) *PSP* values on SYM-PART2 with different population sizes



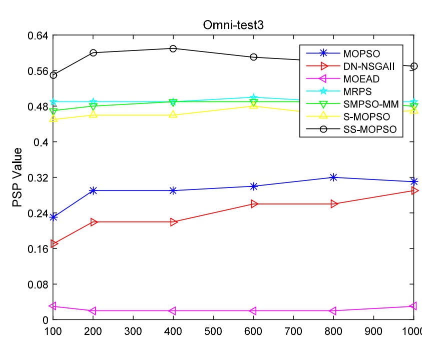
(k) *PSP* values on SYM-PART3 with different population sizes



(l) *PSP* values on Omni-test1 with different population sizes

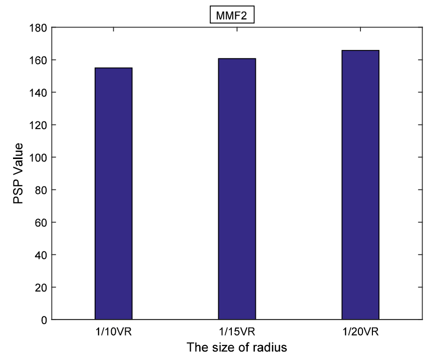


(m) *PSP* values on Omni-test2 with different population sizes

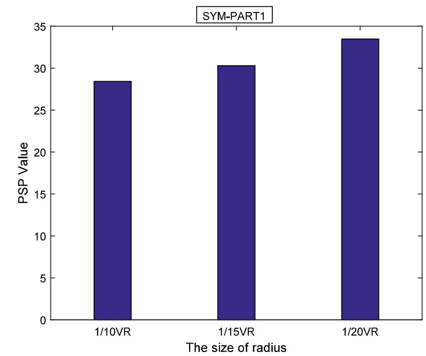


(n) *PSP* values on Omni-test3 with different population sizes

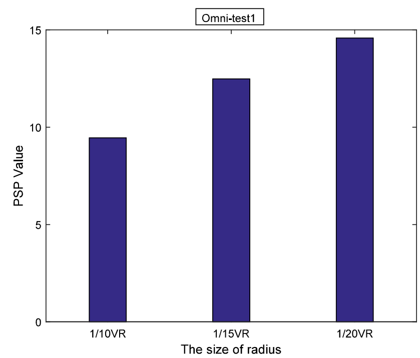
Fig. 9. (*continued*).



(a) *PSP* values on MMF2



(b) *PSP* values on SYM-PART1



(c) *PSP* values on Omni-test1

Fig. 10. PSP values with different radius.

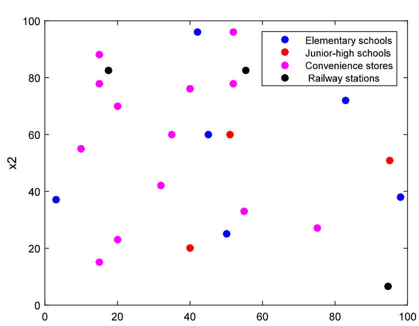


Fig. 11. The map-based four-objective test problem.

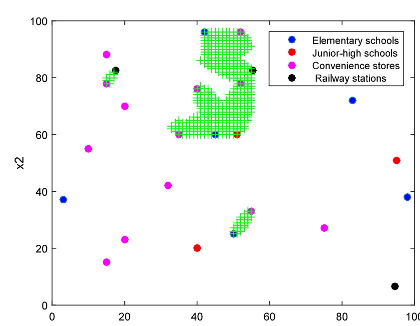
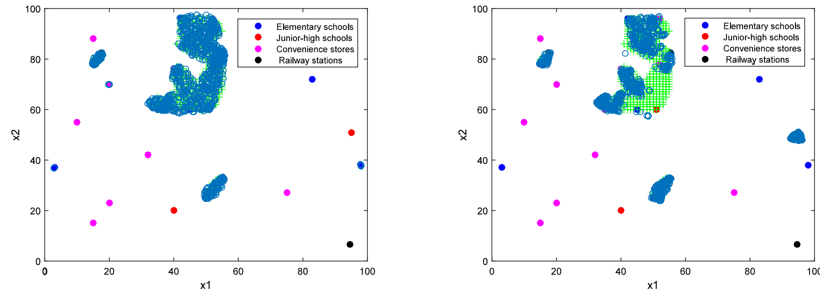


Fig. 12. Pareto regions of the map-based test problem.

5.3. Experimental results with different population sizes

In this section, the seven compared algorithms are used to investigate the effect of different population sizes on the fourteen test functions. The mean *PSP* values obtained by the algorithms for 20 experimental runs are shown in Fig. 9(a)–(n). The figures show that SS-MOPSO realizes the best *PSP* values for a majority of the test functions under different population sizes. The results prove that the SS-MOPSO is competitive for solving multimodal multi-objective problems.

It can be seen from Fig. 9 that the population size affects the performance of the algorithms. The optimal population size for each algorithm is different for the test functions. With respect to MMF1, Fig. 9(a) shows that the *PSP* values obtained by all algorithms increases along with the population size, except for MOEAD and MOPSO. The results obtained by SS-MOPSO, SMPSO-MM, MRPS and S-MOPSO do not change much on MMF2 since the population size 200, and SS-MOPSO, SMPSO-MM and MRPS achieve their best *PSP* values with a population size of 400. However, S-MOPSO gets its largest *PSP* value when the population size is 200. As for MMF3, SS-MOPSO, SMPSO-MM, MRPS and S-MOPSO perform best with a population size of 400. In contrast, DN-NSGAII achieves the best *PSP* values when the population size is 1000. The results in Fig. 9(d)–(h) displays that the *PSP* values obtained by the employed algorithms on MMF4-MMF8 change in a similar trend as that of MMF1. For SYM-PART1, SS-MOPSO achieves the best *PSP* value with a population size of 800, while SMPSO-MM and MRPS both obtain the best *PSP* value with a population size of 600. With respect to the more complex problems, SYM-PART2, SYM-PART3, and Omni-test1-3, SS-MOPSO shows a superior performance with different population size compared with other algorithms as shown in Fig. 9(j)–(n).



(a) Obtain solutions by SS-MOPSO (b) Obtain solutions by DN-NSGAII

Fig. 13. Obtain solutions by the two algorithms.

5.4. Experimental results with different radius

In this paper, the radius is preset to determine the size of the species. Hence, it is understandable that the radius plays an important role on the performance. Generally, the radius of species is set normally to a value between 1/20 to 1/10 of the allowed variable range (VR). In order to verify the effect of the species radius on the performance of the algorithm, different radius are employed for the experiments on MMF2, SYM-PART1, and Omni-test1. The mean *PSP* values of 20 experiments are shown in Fig. 10(a)–(c). From the figures, the mean *PSP* values increase as the radius decreases for all the three test functions. The reason is that as the radius decreases, the number of species formed in the decision space increases. Thus, the algorithm can locate more Pareto optimal solutions and the population diversity in the decision is enhanced. Additionally, the figures shows that the *PSP* values of MMF2 show a smaller change with the increase of the radius compared with those of SYM-PART1 and Omni-test1. The reason for this phenomenon is that the variable range of MMF2 is small, therefore, the change of radius is tiny, which has a little impact on the performance of the algorithm.

5.5. Application

Multimodal multi-objective problem exists widely in the real world. In this section, the proposed algorithm is tested on a real-life problem in order to demonstrate the performance of SS-MOPSO. In Fig. 11, the map-based test problem generated from an actual real-world map is displayed [44]. The test problem involves six elementary schools (blue circles), three junior-high schools (red circles), thirteen convenience stores (pink circles), and three railway stations (black circles), and the location of each school (store, station) is specified according to the actual real-world map. Four objectives to be minimized are defined using the Euclidean distance from a solution x in the decision space as follows:

f_1(x): Distance to the nearest elementary school (blue circle),
f_2(x): Distance to the nearest junior-high school (red circle),
f_3(x): Distance to the nearest convenience store (pink circle),
f_4(x): Distance to the nearest railway station (black circle).

The Pareto optimal regions of the test problem are shown in Fig. 12, and the problem has three disconnected Pareto optimal regions, that is, the three green areas in Fig. 12. The three Pareto optimal regions [44] are depicted by examining the Pareto optimality of each of the 101 × 101 points ($x_1 = 0, 1, 2, \ldots, 100$; $x_2 = 0, 1, 2, \ldots, 100$) in the decision space [0,100] × [0,100].

To demonstrate the effectiveness of SS-MOPSO, it is used to solve the map-based problem. Moreover, DN-NSGAII is also employed to compare. The parameter settings of the two algorithms are as follows. The maximal number of evaluations are set to 80 000 and the population size are set to 1000. In SS-MOPSO, the radius is set to 25. In Fig. 13(a)–(b), the Pareto solution sets obtained by a single run of each algorithm are shown. And we can see that the performance of SS-MOPSO is better than DN-NSGAII. It can be seen from Fig. 13(a) that the SS-MOPSO is almost able to cover all the three Pareto optimal regions of the problem, which proves the effectiveness of SS-MOPSO. However, the PSs obtained by DN-NSGAII is incomplete.

6. Conclusions

In this paper, a self-organized speciation based multi-objective particle swarm optimizer is proposed to solve multimodal multi-objective problems. In SS-MOPSO, the speciation strategy is employed to establish multiple stable niches/subpopulations to optimize towards a large number of Pareto-optimal solutions in parallel. Meanwhile, a self-organized mechanism is proposed to improve the efficiency and the performance of the original speciation strategy.

The proposed algorithm is compared with several state-of-the-art multi-objective or multimodal multi-objective algorithms. All the algorithms are tested on fourteen multimodal multi-objective test functions. Experimental results demonstrated that the proposed SS-MOPSO algorithm shows a better performance and has the ability to retain more and better distributed Pareto optimal solutions in the decision space than other compared algorithms. However, the SS-MOPSO does not perform the best on the multimodal multi-objective test functions with irregular PSs, such as MMF7. In addition, the effect of the species radius on the algorithm performance is also verified and analyzed.

At last, the proposed SS-MOPSO is applied on a practical multimodal multi-objective problem, the map-based test problem. Results show that SS-MOPSO is feasible and effective in solving the problem, and shows a superior performance compared with DN-NSGAII.

In our future work, new multimodal multi-objective test functions with more objectives and larger number of variables will be designed. Moreover, the SS-MOPSO will be improved to solve the problems with different complexity and the more complex practical problems.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105886.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61673404, 61876169, 61873292, 61976237), the Key Scientific Research Project of Colleges and Universities in Henan Province (Grant No. 19A120014) and the Research Award Fund for Outstanding Young Teachers in Henan Provincial Institutions of Higher Education, China (2016GGJS-094).

References

[1] J.J. Liang, C.T. Yue, B.Y. Qu, Multimodal multi-objective optimization: a preliminary study, in: Proc. of IEEE International Conference on Evolutionary Computation, 2016, pp. 2454–2461.

[2] M. Preuss, Multimodal Optimization By Means of Evolutionary Algorithms, Springer International Publishing, 2015.

[3] J. Liang, W.W. Xu, C.T. Yue, K.J. Yu, et al., Multimodal multiobjective optimization with differential evolution, Swarm Evol. Comput. 44 (2019) 1028–1059.

[4] X.D. Li, Niching without niching parameters: Particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14 (1) (2010) 150–169.

[5] X.D. Li, A multimodal particle swarm optimizer based on fitness Euclidean-distance ration, in: Proc. Genetic and Evolutionary Computation Conference, 2007, pp. 78–85.

[6] B.Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (3) (2013) 387–402.

[7] R. Cheng, M. Li, K. Li, X. Yao, Evolutionary multiobjective optimization-based multimodal optimization : fifitness landscape approximation and peak detection, IEEE Trans. Evol. Comput. 22 (5) (2018) 692–706.

[8] Q. Yang, W.N. Chen, Z. Yu, T. Gu, et al., Adaptive multimodal continuous ant colony optimization, IEEE Trans. Evol. Comput. 21 (2) (2017) 191–205.

[9] M. Preuss, Niching methods and multimodal optimization performance, in: Proc. Multimodal Optimization by Means of Evolutionary Algorithms, 2015, pp. 115–137.

[10] P. Haghbayan, H. Nezamabadi-pour, S. Kamyab, A niche GSA method with nearest neighbor scheme for multimodal optimization, Swarm Evol. Comput. 35 (2017) 78–92.

[11] S. Kamyab, M. Eftekhari, Using a self-adaptive neighborhood scheme with crowding replacement memory in genetic algorithm for multimodal optimization, Swarm Evol. Comput. 12 (2013) 1–17.

[12] Q. Ling, G. Wu, Z.Y. Yang, Q.P. Wang, Crowding clustering genetic algorithm for multimodal function optimization, Appl. Soft Comput. 8 (1) (2008) 88–95.

[13] D. Parrott, X.D. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput. 10 (4) (2006) 440–458.

[14] W.J. Luo, X. Lin, T. Zhu, P. Xu, A clonal selection algorithm for dynamic multimodal function optimization, Swarm Evol. Comput. (2018) http://dx.doi.org/10.1016/j.swevo.2018.10.010.

[15] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: Proc. IEEE Congress on Evolutionary Computation, 2004, pp. 1382–1389.

[16] M.Q. Li, D. Lin, J. Kou, A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization, Appl. Soft Comput. 12 (3) (2012) 975–987.

[17] P.S. Oliveto, D. Sudholt, C. Zarges, On the benefits and risks of using fitness sharing for multimodal optimisation, Theoret. Comput. Sci. 773 (2019) 53–70.

[18] C.Y. Lin, W.H. Wu, Niche identification techniques in multimodal genetic search with sharing scheme, Adv. Eng. Softw. 33 (11) (2002) 779–791.

[19] X.D. Li, M.G. Epitropakis, K. Deb, Seeking multiple solutions: an updated survey on niching methods and their applications, IEEE Trans. Evol. Comput. 21 (4) (2017) 518–538.

[20] J.P. Li, M.E. Balazs, G.T. Parks, P.J. Clarkson, A species conserving genetic algorithm for multimodal function optimization, Evol. Comput. 1 (3) (2002) 207–234.

[21] B.Y. Qu, 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.

[22] K. Deb, 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.

[23] A.M. Zhou, Q.F. Zhang, Y.C. 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.

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

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

[26] Y.P. Liu, G.G. Yen, D.W. Gong, A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies, IEEE Trans. Evol. Comput. 23 (4) (2019) 660–674.

[27] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc international conference on neural networks, 1995, pp. 1942–1948.

[28] X.D. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (2011) 210–224.

[29] H. Wang, G.G. Yen, G. Luo, Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system, IEEE Trans. Cybern. 47 (6) (2016) 1446–1459.

[30] P. Danie, X.D. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput. 10 (4) (2006) 440–458.

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

[32] Y.K. Li, Y.L. Chen, J.H. Zhong, Z.X. Huang, Niching particle swarm optimization with equilibrium factor for multi-modal optimization, Inform. Sci. 494 (2019) 233–246.

[33] R. Brits, A.P. Engelbrecht, F. van den Bergh, A niching particle swarm optimizer, in: Proc. Asia-Pacific Conf. Simulated E Learning, 2002, pp. 692–696.

[34] B.Y. Qu, P.N. Suganthan, J.J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.

[35] R. Brits, A.P. Engelbrecht, F. van den Bergh, Solving systems of unconstrained equations using particle swarm optimization, in: Proc. IEEE Congress on Systems, Man, Cybernetics, 2002, pp. 102–107.

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

[37] 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.

[38] X.D. Li, Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization, in: Genetic and Evolutionary Computation Conference, Springer, Berlin, Heidelberg, 2004, pp. 105–116.

[39] G. Rudolph, B. Naujoks, 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.

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

[41] M. Reyes-Sierra, C.A.C. Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, Int. J. Comput. Intell. Res. 2 (3) (2006) 287–308.

[42] Q.F. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.

[43] M. Clerc, J. Kennedy, The particle swarm-explosion stability and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.

[44] H. Ishibuchi, N. Akedo, Y. Nojima, A many-objective test problem for visually examining diversity maintenance behavior in a decision space, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ACM, 2011, pp. 649–656.