IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 20, 2020, accepted October 7, 2020, date of publication October 14, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031002

A Hybrid Leader Selection Strategy for Many-Objective Particle Swarm Optimization

MAN-FAI LEUNG[1], (Member, IEEE), CARLOS ARTEMIO COELLO COELLO[2], (Fellow, IEEE), CHI-CHUNG CHEUNG[3], (Senior Member, IEEE), SIN-CHUN NG[1], (Senior Member, IEEE), AND ANDREW KWOK-FAI LUI[1], (Member, IEEE)

[1]School of Science and Technology, The Open University of Hong Kong, Hong Kong
[2]Departamento de Computación, CINVESTAV-IPN, Mexico City 07360, Mexico
[3]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Man-Fai Leung (mfleung@ouhk.edu.hk)

The work of Carlos Artemio Coello Coello was supported in part by CONACyT under Project 2016-01-1920, and in part by the SEP-Cinvestav under Grant 4.

ABSTRACT Many existing Multi-objective Particle Swarm Optimizers (MOPSOs) may encounter difficulties for a set of good approximated solutions when solving problems with more than three objectives. One possible reason is that the diluted selection pressure causes MOPSOs to fail to generate a set of good approximated Pareto solutions. In this paper, a new approach called the Hybrid Global Leader Selection Strategy (HGLSS) is proposed to deal with many-objective problems more effectively. HGLSS provides two global leader selection mechanisms: one for exploration and one for exploitation. Each particle (solution) can choose one of these two leader selection schemes to identify its global best leader. An external archive is adopted for maintaining the diversity of the found solutions and it contains the final solution reported at the end of the run. The update of the external archive is based on both Pareto dominance and density estimation. The performance of the proposed approach is compared with respect to nine state-of-the-art multi-objective metaheuristics in solving several benchmark problems. Our results indicate that the proposed algorithm generally outperforms the others in terms of Modified Inverted Generational Distance (IGD$^+$) indicator.

INDEX TERMS Many-objective optimization, particle swarm optimization, leader selection.

I. INTRODUCTION

Multi-objective optimization involves optimizing two or more (normally conflicting) objective functions simultaneously and it frequently arises in many application domains such as business and engineering [1]–[3]. In general, minimizing a multi-objective optimization problem (MOP) with K objectives can be stated as:

$$\min\ F(x) = (f_1(x), f_2(x), f_3(x), \ldots, f_K(x)) \tag{1}$$

where $x \in \Re^M$ is an M-dimensional set of decision variables.

A solution is non-dominated if none of the objective functions can be further improved without degrading some of the other objective values. Solution x dominates solution y, denoted by $x \prec y$, if and only if $f_k(x) \leq f_k(y)$ for all $k = 1, 2, \ldots, K$ and k^* exists such that $f_{k*}(x) < f_{k*}(y)$. Also, if no x' exists in the decision space such that $x' \prec x$, x is defined as a Pareto optimal solution (POS). The Pareto set (PS) contains all POSs and its image (i.e., their corresponding objective function values) is called the Pareto front (PF), defined by $\text{PF} = \{F(x)|x \in PS\}$.

Over the past few decades, different Multi-objective evolutionary algorithms (MOEAs) have been proposed and their abilities in solving MOPs with few objectives (two or three) have been shown [4]–[7]. However, several studies [8], [9] have shown that the performance of most MOEAs (particularly those based on Pareto ranking) severely deteriorates when dealing with problems having more than three objectives (they are called many-objective optimization problems (MaOPs)). As the number of objectives increases, the proportion of non-dominated solutions increases sharply [10], and thus the selection pressure provided by the Pareto optimal relation is quickly diluted.

Some MOEAs have recently been proposed to handle MaOPs. From among them, the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [11], [12] is

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed A. Zaki Diab.

the most popular. MOEA/D decomposes a MOP into a set of single-objective sub-problems by using a scalarizing function and these sub-problems are then simultaneously optimized. A set of weight vectors must be assigned properly to get a set of good approximated Pareto solutions. However, some studies have shown that MOEA/D with fixed weight vectors may not be able to approximate the whole Pareto front [13], [14]. Another popular approach is to make use of hypervolume-based MOEAs [15]–[17], but this approach may not be suitable for problems having more than five objectives [18] due to the high computational cost involved in computing exact hypervolume contributions.

Particle swarm optimization (PSO) is a population-based metaheuristic inspired on the flight patterns of a flock of birds [19], [20]. Over the years, a wide variety of Multi-Objective Particle Swarm Optimizers (MOPSOs) have been proposed [21]–[23]. The study from [24] analyzed the performance of six popular MOPSOs over a set of benchmark problems. This study showed that the Speed-constrained MOPSO (SMPSO) [25] was able to outperform other MOPSOs in several MOPs having 2 and 3 objectives, but this approach was not properly tested on MaOPs.

There are, however, several proposals of MOPSOs for properly dealing with MaOPs. For example, Britto and Pozo [26] proposed the use of reference points to update the archive to address the issue of scalability in MOPSOs when solving MaOPs. They showed that the approximated Pareto solutions of the proposed algorithm were close to the selected reference points. On the other hand, Wickramasinghe and Li [27] proposed a user-preference-based MOPSO which does not rely on the use of dominance comparisons, but uses a distance metric as its guidance method. This MOPSO is able to converge close to the preferred regions, but such regions have to be specified by the decision maker beforehand.

As described before, the main difficulty in MOPSOs is the diluted selection pressure [10] that significantly affects performance when the number of objectives increases. This paper develops a new many-objective particle swarm optimizer that can handle the convergence and diversity properly at the same time by using a hybrid global leader selection strategy (HGLSS). Moreover, an external archive is used to maintain both the diversity of the algorithm and the approximated solutions. Our performance investigation shows that the proposed algorithm outperforms other popular multi-objective optimization algorithms in some benchmark many-objective optimization problems in terms of the modified inverted generational distance (IGD$^+$) indicator.

The remainder of this paper is organized as follows. Section II introduces a few preliminary concepts on particle swarm optimization and multi-objective particle swarm optimization. Section III presents our proposed algorithm called MOPSO-HGLSS. Section IV shows the performance investigation in four sub-sections. The first sub-section introduces a performance measure called IGD$^+$ [28] which is used in this paper to compare our results with respect to those of other approaches. The second sub-section discusses the parameters settings in MOPSO-HGLSS. The third sub-section compares our proposed HGLSS with respect to other leader selection strategies under the framework of the MOPSO algorithm. The last sub-section compares the performance of MOPSO-HGLSS with respect to nine popular population-based metaheuristics (SMPSO [25], dMOPSO [29], MOPSOhv [30], MaPSO [31], MOEA/D [32] NSGA-III [33], DBEA [34], RVEA [35] and ARMOEA [36]), in terms of IGD$^+$ with different scalable MOPs (using 3, 5, 8 and 10 objectives). Section V presents our conclusions and some possible paths for future research.

II. PRELIMINARIES

A. PARTICLE SWARM OPTIMIZATION

PSO was originally proposed by James Kennedy and Russell C. Eberhart in 1995 [19]. In PSO, a group of particles (solutions) is randomly initialized within the valid ranges of the decision variables. Then, the velocity of each particle is initialized and the whole swarm starts its motion. At every cycle, the movement of each particle is influenced by its personal best position and the best global position in the swarm. Let x_i^t be the position of the i^{th} particle at cycle t, its velocity v_i^t is updated as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (x_{pb,i} - x_i^t) + c_2 r_2 (x_{gb} - x_i^t) \tag{2}$$

where ω is the inertia weight; c_1 and c_2 are defined as constants representing the cognitive and social factors, respectively; r_1 and r_2 are two random (continuous) variables defined within the range [0, 1]; $x_{pb,i}$ is the personal best position of the i^{th} particle and x_{gb} is the global best position in the swarm. The personal best $x_{pb,i}$ of i^{th} particle is replaced by its new particle if its current fitness value is better, i.e., if $f(x_i^t) < f(x_{pb,i})$, then $x_{pb,i} = x_i^t$. The global best of the swarm x_{gb} is identified by finding the one with the smallest fitness value, i.e., $x_{gb} = \arg\min f(x_i^t)$ for all i.

The position of the i^{th} particle is updated by using the following equation:

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{3}$$

B. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

In the standard PSO, the whole swarm tends to converge to the global best leader because all particles in the swarm share a common global best leader, but some modifications are required for PSO to solve MOPs. The first modification is an external archive, which is widely accepted for use with a (pre-defined) fixed size as a means to store the non-dominated solutions generated by the algorithm. Additionally, the external archive should maintain a set of good non-dominated solutions in terms of diversity and convergence. The second modification is a global leader selection scheme, which can be used to identify the global best leader for each particle from the external archive. This mechanism is very important in MOPSOs because it affects the flight trajectories of particles and hence affects their convergence and diversity. Due to the importance of the leader selection scheme, a variety of proposals are currently available (see [37]–[40]). Among them, WSum [41], NWSum [42] and crowding distance [37] are the most popular ones. The scheme of WSum was introduced for personal best selection by assigning a higher weight to those criteria in which particle is already relatively good. In [42], the author considered this scheme for global best selection. For a particular M-dimensional particle $x \in \Re^M$ with an archive member $y \in Y$ where Y is the set of all members in the archive, the weighted sum value is calculated as follows and the archive member with the smallest weighted sum value is chosen as the global leader of that particle.

$$\mathrm{WSum}(y, x) = \sum_{k=1}^{K} \frac{f_k(x)}{\sum_j^J f_j(x)} f_k(y) \tag{4}$$

The scheme of NWSum is another version of WSum whose particles identify their global best leaders using (4) but with the maximum weighted sum value. The scheme of Crowding Distance (CD) was proposed in [43] to estimate the density of solutions and they are assigned with a CD value. The boundary solutions are assigned with infinity CD values. Then, the solutions are sorted according to the CD values in descending order. The top 5% of the sorted solutions will be randomly selected as leaders. The aforementioned strategies are also applied to select the personal best in MOPSOs [42]. Besides, various other personal best selection strategies are proposed and some of them are widely used. In [44], the personal best is randomly selected from an external archive. This approach is computationally efficient and favors diversity. However, it may lead to a lack of convergence [42]. In [29], [45], the personal best of each particle is updated if the new aggregation value is better. Furthermore, mutation is widely used to increase the exploratory capability of MOPSOs and to prevent premature convergence.

III. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION WITH HYBRID GLOBAL LEADER SELECTION STRATEGY

A. MOTIVATION

As discussed previously, the global leader selection scheme plays an important role which affects both the convergence and the diversity of MOPSO and it is difficult to maintain both when using a single global leader selection scheme, especially when the number of objectives is large. Thus we propose using a Hybrid Global Leader Selection Scheme (HGLSS). Under our proposed scheme, two global leader selection schemes are available for every particle to choose: one is for exploration and the other one is for exploitation. Each particle recognizes its global best based on one of these two leader selection schemes. In this work, we propose the use of two existing leader selection strategies for particles. They are Euclidean Distance Strategy (EDS) [46] and Space Expanding Strategy (SES) [47].

For exploitation, EDS is adopted to guide particles to the closest archive members so that particles can reach their leaders within a small number of generations. Under this scheme, particles select their own global leaders from the external archive (this will be explained in Section III-B) based on the Euclidean distance calculation. For K objectives, the Euclidean distance (ED) of two solutions x_1 and x_2 is given by:

$$\mathrm{ED}(x_1, x_2) = \sqrt{\sum_{k=1}^{K} (f_k(x_1) - f_k(x_2))^2} \tag{5}$$

To determine the global leader of a particle, the Euclidean distance between the particle and all members in the external archive is calculated and the archive member with the shortest Euclidean distance (in objective space) is chosen as the global leader of that particle, i.e.,

$$\mathrm{ED}(y^*, x) \leq \mathrm{ED}(y, x) \tag{6}$$

given that $y = y_1, y_2, \ldots, y_n \in Y$ where Y is the set of all members in the archive and y^* is the global leader of a particle where $y^* \in Y$.

For exploration, SES is responsible for maintaining the diversity of the MOPSO. SES attempts to push some particles to the boundary leaders and new solutions are aimed to be discovered close to the boundary leaders. In SES, the objective to select a leader is random at the beginning. Then, all external archive members are sorted according to their fitness values using the selected objective. Finally, one archive member is randomly selected from the top 5% of the sorted archive [48], [49] and the selected member becomes the global leader of the particle. This 5% is to weight the focus away from the compromise solutions. Under SES, particles in the swarm are randomly pushed toward to the boundary of the external archive so that the spread of the particles can be increased. Algorithm 1 shows the pseudocode of SES.

Algorithm 1 Pseudocode of SES

Input: External archive Y

1. Create a variable *objIndex* for holding the objective index;
2. Randomly select one of the objectives, save the objective index to *objIndex*;
3. Sort the external archive members according to objective *objIndex* in ascending order of objective values;
4. **for** each particle i **do**
5. Randomly select a member from the top 5% of the sorted archive as the global best x_{gb};
6. **end for**

Output: A global best x_{gb}

In our proposed schemes, particles select their leaders from EDS or SES. The selection is based on the probability ρ: A random number between 0 and 1 is generated. If the random number is less than ρ, SES is selected; otherwise EDS is selected. To maintain the ability of exploitation, ρ cannot be too large (e.g., 0.1); otherwise, there may not be enough particles to get enough solutions for achieving convergence. Due to the importance of ρ, the effect of this parameter on the overall performance will be investigated in Section IV-B.

B. EXTERNAL ARCHIVE

An external archive is required in MOPSO-HGLSS to store a set of good approximated Pareto solutions. The external archive has four main duties: 1) store new non-dominated solutions; 2) remove solutions if they become dominated; 3) select and remove archive members if the archive is full; and 4) maintain the diversity of archive members. In MOPSO-HGLSS, two mechanisms have been designed to achieve these duties: 1) Pareto Dominance Selection (PDS) (for the first two duties), and 2) Neighbor Factor Selection (NFS) (for the last two duties).

Algorithm 2 Pseudocode of the Update of the External Archive

Input: Particles at cycle t, i.e., x_i^t for all i; the external archive Y^*

for each particle x_i^t **do**
  insert_x=0;
  for each archive member $y_j \in Y^*$ **do**
    if $x_i^t \prec y_j$ **then**
      Remove y_j from Y^*;
      insert_x=1;
    else if $x_i^t \nprec y_j$ **then**
      if archive Y^* is not full **then**
        insert_x=1;
      else
        Calculate the NFS values for each $y_j \in Y^*$;
        Set the NFS values of boundary archive members to infinity;
        Remove the archive member with the smallest NFS value;
        insert_x=1;
      end if
    end if
  end for
  if insert_x==1 **then**
    Insert x_i^t to Y^*;
  end if
end for

Output: Y^*

PDS is widely adopted in multi-objective evolutionary algorithms (see e.g., [50]): if a new solution dominates at least one archive member, such archive member(s) will be removed and the new solution will be inserted into the archive; otherwise, the new solution will be discarded. If the new solution is incomparable ($s1$ and $s2$ are incomparable if neither $s1 \nprec s2$ nor $s2 \nprec s1$, and $s1 \neq s2$) with the archive members (i.e., all archive members and the new solution are non-dominated solutions) and the archive is not full, the new solution will be added into the archive. However, if the new solution is incomparable with the archive members and the archive is full, an additional criteria which estimates the density of the archive members based on the Euclidean distance [42] (we call it NFS in this paper) is used to determine which solution should be removed from the archive. Note that the role of NFS is to maintain the diversity of archive members. If b is the new solution, NFS removes a solution with the smallest NF (Neighbor Factor) value among all solutions where the NF value of a solution is defined as follows:

$$\mathrm{NF}(y) = \mathrm{ED}(y1, y) + \mathrm{ED}(y2, y) \tag{7}$$

where $\mathrm{ED}(y1, y) \leq \mathrm{ED}(y2, y) \leq \mathrm{ED}(y', y)$ for $y, y1, y2 \in Y + b$ and $y' \in Y - y1, y2$. The NF value of a solution indicates the diversity of a solution among all archive members. If the NF value of a solution is small, it means it is close to its neighbors (i.e., $y1$ and $y2$) and thus the diversity is not good. If this solution is removed, the diversity of all archive members will increase. Note that the boundary members (i.e., solutions with the smallest fitness value(s) in one or more objectives) will not be considered for removal because they need to remain in the archive to maintain a well-distributed Pareto front.

C. MUTATION OPERATOR

Mutation operators are widely used in MOPSOs to prevent premature convergence and increase their exploratory capabilities. Note that the mutation rate should not be too high; otherwise, the overall performance of a MOPSO will degrade. Our proposed algorithm adopts polynomial-based mutation [51]. Let r be a random number uniformly distributed in [0, 1] and η_m be the index for polynomial-based mutation, then, the i^{th} particle to be mutated is calculated at iteration t as follows:

$$x_i'^t = x_i^t + \eta_q(\bar{x} - \underline{x}) \tag{8}$$

where $\bar{x}, \underline{x}$ are the upper and lower bounds of x_i^t, η_q is defined as:

$$\begin{cases} [2r + (1-2r)(1-\eta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1, & r \leq 0.5 \\ 1 - [2(1-r) + 2(r-0.5)(1-\eta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}, & r > 0.5, \end{cases}$$

and

$$\eta_1 = \frac{x_i^t - \underline{x}}{\bar{x} - \underline{x}}, \quad \eta_2 = \frac{\bar{x} - x_i^t}{\bar{x} - \underline{x}}.$$

D. THE FULL MOPSO-HGLSS ALGORITHM

Algorithm 3 shows the pseudocode of the full MOPSO-HGLSS algorithm (i.e., HGLSS is implemented in the MOPSO algorithm). At the beginning, the external archive is initialized along with the position, speed and best locations of all particles. Then, the swarm is evaluated and the archive is updated. Non-dominated particles will be added into the archive. If the external archive is full, an archive member or a non-dominated particle will be removed based on their NF values. The following procedure will be repeated until the maximum number of generations is reached: particles update their velocities based on their global best and their personal best values. The global best of each particle is determined by using HGLSS while the personal best is updated if it is dominated by the current position. If they are non-dominated to each other, one of them will be randomly selected [42].

Algorithm 3 Pseudocode of the MOPSO-HGLSS Algorithm

Input: An empty external archive $Y = []$
Set the iteration counter $t = 1$;
for each particle i **do**
  Initialize the position randomly within its allowable boundaries;
  Initialize personal best $x_{pb,i} = x_i^t$;
  Initialize its speed to zero $v_i^t = 0$;
  Evaluation;
end for
Update the external archive Y;
for $t = 2$ to the last generation **do**
  for each particle i **do**
    Use HGLSS to generate x_{gb};
    if $x_i^t \prec x_{pb,i}$ **then**
      $x_{pb,i} = x_i^t$;
    else if rand > 0.5 **then**
      $x_{pb,i} = x_i^t$;
    end if
    Update the velocity using (2);
    Update the position using (3);
  end for
  Mutation;
  Boundary;
  Evaluation;
  Update the external archive Y;
  $t = t + 1$;
end for
Output: Y^*

After updating the position of all particles, mutation occurs with a designated probability to enhance the exploratory ability of the proposed algorithm. To ensure particles moving within the search space, bounds checking for all the particles is conducted after updating the positions. If the decision variable of a particle is smaller than the lower bound or larger than the upper bound, then it will be assigned to the lower or upper bound value. Finally, the evaluation of all particles is carried out. Before proceeding to the next generation, the external archive and the personal best of all particles are updated.

IV. PERFORMANCE COMPARISONS

This section introduces the performance measures used in the performance comparison, investigates the parameters settings of HGLSS, and compares HGLSS with respect to other popular algorithms.

A. PERFORMANCE MEASURES

As the convergence and the diversity of approximated Pareto solutions are two main issues in multi-objective optimization, we decided to adopt the Modified Inverted Generational Distance (IGD^+) indicator [28].

The idea of using an inverted form of the Generational Distance indicator was apparently proposed first by Bosman and Thierens [52], although it was first used with the name of Inverted Generational Distance (IGD) in [53]. IGD is able to measure both the convergence and the diversity of the approximated Pareto solutions. Let PF^* be a set of uniformly distributed points sampled from the true Pareto front and Y^* be the set of approximated Pareto solutions. Note that $Y = Y^*$ at the end of the generations. IGD is defined as:

$$\text{IGD}(Y^*, PF^*) = \frac{\sum_{x \in PF^*} \text{ED}(b^*, x)}{|PF^*|} \tag{9}$$

where $\text{ED}(b^*, x) \leq \text{ED}(y, x)$ for $b^* \in Y^*$ and all $y \in Y^*$ and $|PF^*|$ is the cardinality of PF^*. The lower the IGD value is, the better the approximated Pareto solutions will be.

IGD is one of the most popular indicators used for assessing performance of multi-objective optimization algorithms (see e.g., [33], [54]. However, IGD is Pareto non-compliant [55], [56], which may cause misleading results. To make IGD weakly Pareto compliant, the authors in [28] suggested taking the Pareto dominance relation between the approximated solutions and a reference set into account. This means that if an approximated solution is dominated by a reference point, the Euclidean distance is adopted in (9). If they are non-dominated with respect to each other, the minimum distance from the reference point to the region that is dominated by the solution is calculated. The authors also showed that the modified IGD is weakly Pareto compliant. For each test instance in this paper, PF^* contains 100,000 Pareto optimal points which were generated uniformly for calculating the IGD^+ value of the solutions generated by the algorithms under evaluation.

B. PERFORMANCE INVESTIGATION OF HGLSS

This sub-section investigates the effect of ρ on the performance of HGLSS in some optimization problems where ρ is the probability that particles identify their global best using either EDS or SES (i.e., our proposed HGLSS). DTLZ1 and DTLZ2 are used: DTLZ1 has many local Pareto fronts and it is used to test whether an algorithm can converge into the true Pareto front, while DTLZ2 is used to investigate whether an algorithm can maintain a good solution distribution. In addition, WFG6, WFG7, WFG8 and WFG9 are also used. All these tests were conducted using the mentioned test problems with ten objectives. Each test was assigned a specified value of ρ.

For each test, 30 independent runs were conducted. The total number of evaluations was set to 80,000 for the DTLZ test problems and 150,000 for the WFG test problems. The size of the swarm and the external archive were set to 100. The values of c_1 and c_2 were set to 2.5. The inertia weight was set to 0.1, $\eta_m = 20$ and the mutation rate p_m was set to $1/n$, where n is the number of decision variables.

Figure 1 shows the change of the mean IGD^+ value for the obtained solutions when the value of ρ changes from 0 to 1. The figure shows that the mean IGD^+ value of the approximated Pareto optimal solutions for each chosen test problem is the lowest when $\rho = 0.1$. The result meets

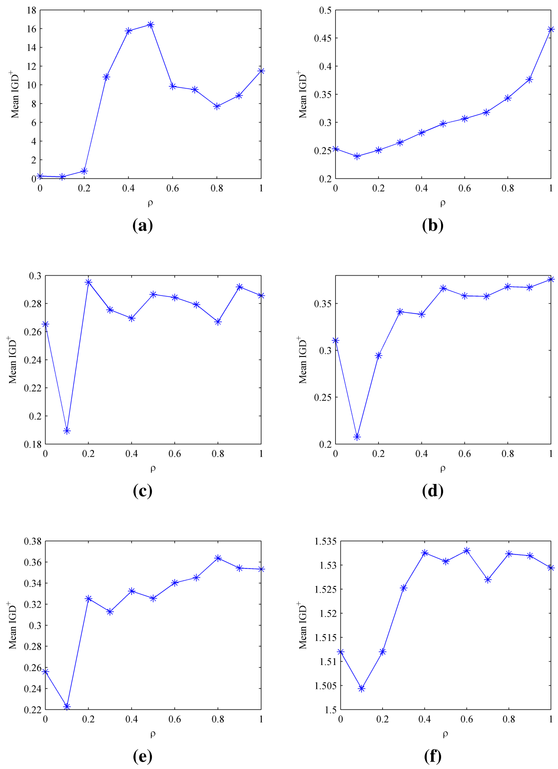


FIGURE 1. The mean IGD$^+$ value vs. ρ for (a) DTLZ1, (b) DTLZ2, (c) WFG6, (d) WFG7, (e) WFG8 and (f) WFG9 with ten objectives.

the expectation discussed in Section III-A (i.e., EDS should dominate SES). Otherwise, particles may possibly disturb their original flights and the swarm may lose exploitation ability for a large value of ρ. As ρ increases, the mean IGD$^+$ value increases, which implies that the performance of the convergence and the diversity of the algorithm get worse.

Based on the performance investigation of the mean IGD$^+$ values on the selected test problems obtained by our proposed algorithm with different probabilities, $\rho = 0.1$ is an appropriate setting for HGLSS and it will be used later on for the rest of our experiments.

C. PERFORMANCE COMPARISONS AMONG HGLSS AND OTHER STATE-OF-THE-ART LEADER SELECTION STRATEGIES

This sub-section compares the performance of HGLSS with five leader selection strategies by using 11 scalable test problems with different numbers of objectives. The five strategies are crowding distance (CD) [37], WSum [41], NWSum [42], SES only and EDS only (both SES and EDS are mentioned in Section III-A). All these leader selection strategies are implemented under the framework of the MOPSO algorithm. Thus six such MOPSO algorithms are called MOPSO-HGLSS, MOPSO-CD, MOPSO-Wsum, MOPSO-NWSum MOPSO-SES and MOPSO-EDS, respectively. For each of the compared algorithms, the size of the swarm and the external archive are all set to 100. The values of c_1 and c_2 are both set to 2.5. The inertia weight is set to 0.1, $\eta_m = 20$ and the mutation rate $p_m = 1/n$. The 11 test problems are (a) DTLZ1 and DTLZ2 from the DTLZ test suite [57], and (b) WFG1 to WFG9 from the WFG test suite [58]. Note that DTLZX-Y refers to the DTLZX test problem with Y objectives. For example, DTLZ1-10 refers to DTLZ1 with ten objectives. For all instances, 30 independent runs are conducted. The maximum number of evaluations of each algorithm is set to 80,000 for the DTLZ test problems and to 150,000 for the WFG test problems. Table 1 shows the mean (outside the parentheses) and the standard deviation (inside the parentheses) of the algorithms in terms of the mean IGD$^+$ value for different MOPs with different numbers of objectives. The best mean is shown in **boldface**. Wilcoxon rank-sum test [59] at a 0.05 significance level was conducted between the proposed algorithm and the five other MOPSOs (with different leader selection schemes), respectively. In Table 1, ‡, † and = are marked next to the values of an algorithm in the tables to denote that the performance of the algorithm is significantly better, worse or has no significant difference with respect to that of MOPSO-HGLSS. Table 2 summarizes the results of the test in terms of IGD$^+$. With respect to IGD$^+$, MOPSO-HGLSS obtained better results in 191 out of 220 performance comparisons. From the results, we conclude that HGLSS performs better that the other compared leader selection strategies with regards to exploration and exploitation under the framework of MOPSO.

D. PERFORMANCE COMPARISONS OF HGLSS-MOPSO WITH OTHER ALGORITHMS

This sub-section compares the performance of HGLSS-MOPSO with respect to that of nine popular multi-objective/many-objective algorithms using 19 scalable test problems with different numbers of objectives (DTLZ1, DTLZ2, WFG1 to WFG9, and MaF1 to MaF8 [60]). For all instances, 30 independent runs were conducted. The maximum number of evaluations of each algorithm was set to 80,000 for the DTLZ test problems, 150,000 for the WFG and MaF test problems.

The compared algorithms can be classified into two groups: MOPSOs and MOEAs. The group of MOPSOs consists of SMPSO [25], dMOPSO [29], MOPSOhv [30], MaPSO [31] and the proposed algorithm, while the other group consists of MOEA/D [32], NSGA-III [33], DBEA [34], RVEA [35] and ARMOEA [36]. In [25], the authors found that the speed of the particles in MOPSO was sometimes too high, making the particles move directly towards the boundaries. To tackle this problem, the authors presented a modified MOPSO algorithm called Speed-constrained Multi-objective PSO (SMPSO) that limits the velocities of the particles. dMOPSO uses decomposition to select leaders and update the external archive. In [30], the authors proposed a hypervolume-based MOPSO called MOPSOhv. This algorithm uses the hypervolume contribution of the archived solutions for selecting the global best and the

TABLE 1. Performance comparisons of MOPSOs with different leader selection strategies in terms of the mean IGD$^+$ value.

	M	MOPSO-HGLSS	MOPSO-CD	MOPSO-WSum	MOPSO-NWSum	MOPSO-SES	MOPSO-EDS
WFG1	3	4.73E-1(3.8E-2)	5.21E-1(1.2E-2)†	4.85E-1(1.0E-2)=	4.79E-1(8.9E-3)=	4.79E-1(6.2E-3)=	**4.59E-1**(2.3E-2)‡
	5	**5.52E-1**(3.6E-2)	6.58E-1(9.7E-3)†	6.34E-1(1.3E-2)†	6.12E-1(1.6E-2)†	6.27E-1(2.6E-3)†	5.60E-1(6.8E-3)=
	8	1.31E+0(3.7E-2)	1.60E+0(7.2E-3)†	1.38E+0(8.2E-3)=	1.32E+0(1.5E-2)=	1.36E+0(6.2E-3)=	**1.29E+0**(3.8E-2)=
	10	3.53E+0(5.9E-2)	4.16E+0(1.9E-2)†	4.11E+0(8.5E-3)†	4.08E+0(9.2E-3)†	4.05E+0(2.8E-2)†	**3.50E+0**(8.8E-2)=
WFG2	3	**2.22E-2**(3.3E-3)	2.01E-1(7.3E-3)†	1.56E-1(4.6E-3)†	2.10E-1(9.8E-3)†	2.52E-2(4.1E-3)†	3.12E-2(6.6E-3)†
	5	**5.41E-2**(5.9E-3)	2.42E-1(6.1E-3)†	2.08E-1(6.9E-3)†	2.10E-1(8.2E-3)†	7.18E-2(7.4E-3)†	9.78E-2(5.4E-3)†
	8	**7.78E-2**(8.2E-3)	4.55E-1(3.2E-2)†	4.11E-1(4.8E-2)†	3.67E-1(6.1E-2)†	1.63E-1(1.2E-2)†	3.46E-1(3.9E-2)†
	10	**1.36E-1**(2.9E-2)	1.81E+0(1.2E-1)†	1.37E+0(2.8E-1)†	1.28E+0(1.7E-1)†	4.10E-1(3.7E-2)†	1.44E+0(1.9E-1)†
WFG3	3	**3.87E-2**(4.8E-3)	3.00E-1(2.4E-2)†	2.37E-1(4.7E-2)†	2.82E-1(4.3E-2)†	8.67E-2(6.8E-3)†	1.03E-1(1.8E-2)†
	5	**2.28E-1**(4.8E-2)	1.65E+0(2.6E-1)†	9.65E-1(1.3E-1)†	1.32E+0(2.1E-1)†	5.18E-1(5.6E-2)†	4.51E-1(7.9E-2)†
	8	**4.60E-1**(6.3E-2)	1.34E+1(2.2E+0)†	1.02E+1(1.1E+0)†	1.21E+1(3.1E+0)†	3.27E+0(1.0E+0)†	4.18E+0(1.3E+0)†
	10	**5.51E-1**(8.2E-1)	4.93E+1(6.0E+0)†	3.99E+1(7.1E+0)†	4.01E+1(1.1E+1)†	1.91E+1(5.6E+0)†	2.16E+1(5.1E+0)†
WFG4	3	**3.01E-2**(5.8E-2)	1.23E-1(1.2E-2)†	1.25E-1(2.0E-2)†	1.17E-1(1.9E-2)†	1.01E-1(1.7E-2)†	6.26E-2(9.8E-3)†
	5	**9.98E-2**(8.3E-2)	2.59E-1(3.5E-2)†	2.63E-1(3.9E-2)†	2.33E-1(5.1E-2)†	3.55E-1(4.8E-2)†	1.99E-1(2.8E-2)†
	8	**1.26E-1**(1.0E-2)	4.63E-1(7.8E-2)†	4.13E-1(7.8E-2)†	4.37E-1(7.3E-2)†	6.59E-1(3.2E-2)†	4.30E-1(6.9E-2)†
	10	**1.51E-1**(1.3E-2)	7.65E-1(9.9E-2)†	6.17E-1(1.3E-1)†	6.02E-1(7.1E-2)†	9.13E-1(7.9E-2)†	6.88E-1(8.2E-2)†
WFG5	3	**4.38E-2**(1.8E-3)	7.14E-2(7.9E-3)†	9.65E-2(6.6E-3)†	9.88E-1(1.0E-2)†	4.86E-2(1.1E-3)†	6.07E-2(4.6E-3)†
	5	**1.11E-1**(4.9E-3)	1.53E-1(8.6E-3)†	1.50E-1(7.1E-3)†	1.58E-1(7.9E-3)†	1.16E-1(5.1E-3)=	1.39E-1(7.1E-3)†
	8	**1.53E-1**(6.9E-3)	2.22E-1(1.9E-2)†	2.12E-1(1.7E-2)†	2.09E-1(3.1E-2)†	1.59E-1(8.1E-3)=	2.00E-1(2.0E-2)†
	10	**1.91E-1**(9.1E-3)	2.51E-1(3.4E-2)†	2.36E-1(4.1E-2)†	2.34E-1(1.8E-2)†	1.98E-1(1.1E-2)=	2.04E-1(2.3E-2)†
WFG6	3	**5.72E-2**(2.1E-3)	6.53E-2(4.1E-3)†	5.80E-2(1.2E-3)=	5.76E-2(1.8E-3)=	5.74E-2(1.7E-3)=	5.74E-2(1.4E-3)=
	5	1.20E-1(1.1E-2)	1.74E-1(2.8E-2)†	1.39E-1(1.1E-2)†	1.43E-1(1.6E-2)†	**1.18E-1**(9.3E-3)=	1.20E-1(1.3E-2)=
	8	**1.52E-1**(9.6E-3)	2.63E-1(2.9E-2)†	2.53E-1(4.3E-2)†	2.08E-1(1.1E-2)†	2.15E-1(4.6E-2)†	2.10E-1(6.1E-2)†
	10	**1.66E-1**(9.8E-3)	2.99E-1(3.7E-2)†	2.62E-1(2.1E-2)†	2.29E-1(2.0E-2)†	2.46E-1(1.6E-2)†	2.35E-1(2.5E-2)†
WFG7	3	**2.88E-2**(8.7E-4)	1.66E-1(1.1E-2)†	1.73E-1(1.9E-2)†	1.48E-1(1.2E-2)†	7.01E-2(1.6E-2)†	6.00E-2(5.6E-3)†
	5	**9.92E-2**(8.3E-3)	2.88E-1(2.1E-2)†	2.79E-1(2.3E-2)†	2.54E-1(2.7E-2)†	2.11E-1(6.8E-3)†	1.84E-1(9.9E-3)†
	8	**1.66E-1**(9.8E-3)	4.01E-1(2.9E-2)†	3.68E-1(3.3E-2)†	3.58E-1(3.1E-2)†	3.00E-1(2.1E-2)†	2.21E-1(1.8E-2)†
	10	**1.78E-1**(9.3E-3)	4.99E-1(3.1E-2)†	4.68E-1(3.7E-2)†	4.36E-1(2.7E-2)†	3.52E-1(1.9E-2)†	2.51E-1(1.6E-2)†
WFG8	3	**6.48E-2**(4.1E-3)	1.32E-1(1.3E-2)†	1.75E-1(6.8E-3)†	1.69E-1(7.1E-3)†	1.01E-1(1.1E-2)†	9.76E-2(6.5E-3)†
	5	**1.18E-1**(9.8E-3)	1.85E-1(1.9E-2)†	1.63E-1(7.9E-3)†	1.59E-1(6.5E-3)†	2.03E-1(8.4E-3)†	1.47E-1(9.1E-3)†
	8	**1.71E-1**(7.6E-3)	2.35E-1(2.3E-2)†	2.21E-1(1.9E-2)†	2.24E-1(1.8E-2)†	2.81E-1(7.3E-3)†	2.12E-1(5.5E-3)†
	10	**2.02E-1**(9.2E-3)	2.91E-1(2.1E-2)†	2.77E-1(3.6E-2)†	2.80E-1(2.0E-2)†	3.38E-1(1.5E-2)†	2.48E-1(1.3E-2)†
WFG9	3	4.39E-2(3.1E-3)	5.28E-2(1.1E-2)†	**3.76E-2**(1.2E-3)‡	4.23E-2(2.8E-3)=	4.40E-2(2.6E-3)=	4.37E-2(1.7E-3)=
	5	1.08E-1(9.8E-3)	1.76E-1(4.8E-2)†	1.26E-1(1.4E-2)†	1.24E-1(1.1E-2)†	**1.05E-1**(8.5E-3)=	1.07E-1(9.1E-3)=
	8	5.46E-1(2.1E-2)	9.01E-1(2.7E-1)†	6.04E-1(5.2E-2)†	6.48E-1(2.1E-1)†	**5.43E-1**(2.3E-2)=	5.48E-1(3.1E-2)=
	10	1.38E+0(1.5E-2)	2.68E+0(9.5E-1)†	1.56E+0(6.8E-1)†	1.60E+0(8.1E-1)†	**1.35E+0**(1.8E-2)=	1.38E+0(1.2E-2)=
DTLZ1	3	**3.29E-2**(7.9E-4)	3.51E+1(1.4E+1)†	3.89E+1(1.6E+1)†	4.11E+1(1.0E+1)†	1.20E+0(4.8E+0)†	5.81E-2(5.7E-3)†
	5	**9.13E-2**(3.4E-3)	3.57E+1(1.2E+1)†	4.11E+1(1.0E+1)†	3.68E+1(1.1E+1)†	5.21E+0(7.0E+0)†	1.21E-1(1.3E-2)†
	8	**1.54E-1**(6.5E-3)	3.66E+1(1.1E+1)†	4.18E+1(1.1E+1)†	3.43E+1(1.2E+1)†	9.28E+0(9.1E+0)†	2.27E-1(1.1E-1)†
	10	**1.86E-1**(1.2E-2)	4.10E+1(1.1E+1)†	4.32E+1(7.4E+0)†	3.42E+1(1.2E+1)†	7.71E+0(1.0E+1)†	2.87E-1(3.5E-2)†
DTLZ2	3	**2.81E-2**(3.1E-3)	7.58E-2(7.9E-3)†	1.22E-1(2.1E-2)†	9.79E-2(3.1E-2)†	4.82E-2(3.0E-3)†	3.51E-2(2.8E-3)†
	5	**1.35E-1**(9.1E-2)	3.01E-1(1.4E-2)†	3.07E-1(1.6E-2)†	3.13E-1(2.8E-2)†	2.54E-1(3.7E-2)†	1.53E-1(2.3E-2)†
	8	**2.51E-1**(2.1E-2)	4.99E-1(2.1E-2)†	5.38E-1(2.3E-2)†	4.81E-1(3.1E-2)†	4.26E-1(2.6E-2)†	3.04E-1(2.9E-2)†
	10	**3.49E-1**(1.9E-2)	5.71E-1(2.6E-2)†	5.99E-1(3.1E-2)†	5.94E-1(3.6E-2)†	5.68E-1(2.7E-2)†	3.91E-1(2.1E-2)†

TABLE 2. Summary of the Wilcoxon rank-sum test results for the selected leader selection strategies with respect to the mean IGD$^+$ value.

	MOPSO-CD	MOPSO-WSum	MOPSO-NWSum	MOPSO-SES	MOPSO-EDS
†	44	40	40	33	34
‡	0	1	0	0	1
=	0	3	4	11	9

†, ‡ and = denote the number of times the performance of the corresponding algorithm is significantly better, worse or has no significant difference with respect to that of the proposed algorithm, respectively.

personal best for every particle in the swarm. In [31], the particle swarm optimizer with the use of scalar projections, is extended for many-objective optimization. MOEA/D is a popular decomposition-based MOEA proposed by

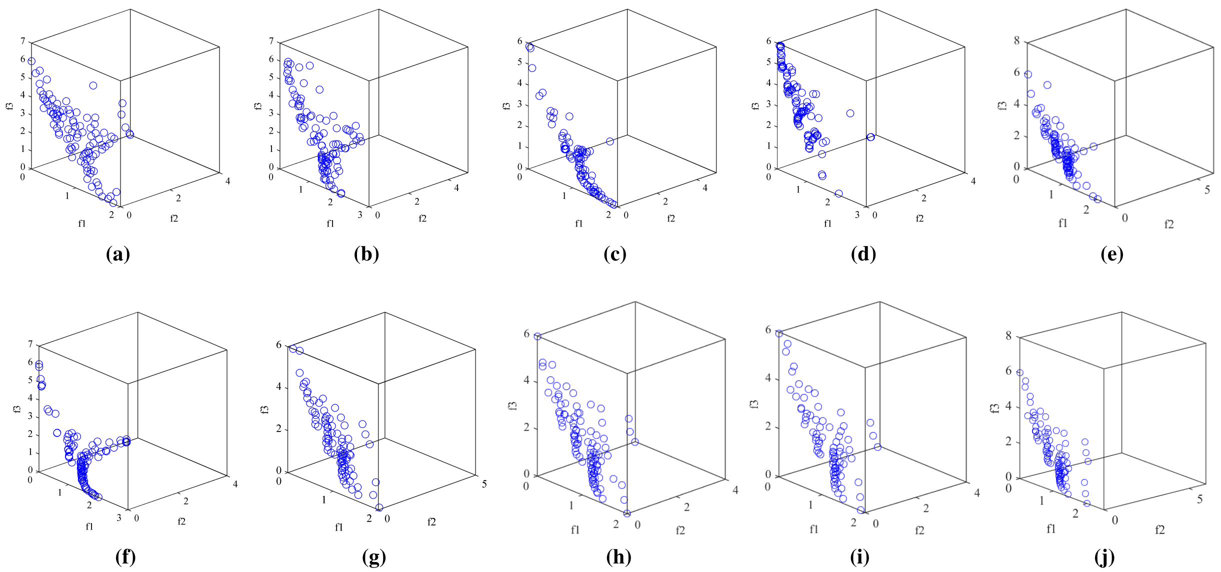


FIGURE 2. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG2-3, respectively.

Li and Zhang [32]. In MOEA/D, a MOP is decomposed into a set of single-objective problems through the use of a scalarizing function, and these sub-problems are simultaneously optimized using neighborhood search. NSGA-III, DBEA and RVEA are decomposition-based algorithms which deal with many-objective problems, whereas ARMOEA is an indicator-based algorithm for many-objective optimization.

For SMPSO, both the swarm size and the archive size are set to 100. The inertia weight is set to 0.1. The values of c_1 and c_2 are assigned randomly between 1.5 and 2.0. The mutation rate is set to $1/n$. For dMOPSO, its inertia weight is assigned randomly between 0.1 and 0.4. The values of c_1 and c_2 are assigned randomly between 1.5 and 2.0. The swarm size is set to 100 and the age threshold is set to 2. For MOPSOhv, both the swarm size and the archive size are set to 100. The inertia weight is set to 0.4. The values of c_1 and c_2 are set to 1.0. The mutation rate is set to 0.5. MOEA/D uses the differential evolution crossover. The crossover probability and the differential weight are set to 1.0 and 0.5, respectively. The neighborhood size is set to 20. The population size of NSGA-III, DBEA and RVEA are set to 105. The population size of MaPSO is set to 92. For ARMOEA, the population size is set to 105. In HGLSS, the size of the swarm and external archive are set to 100. The values of c_1 and c_2 are set to 2.5. The inertia weight is set to 0.1. The mutation rate is set to $1/n$ and its distribution index is set to 20.

Figures 2 to 7 show some approximated Pareto fronts of the median run produced by the compared algorithms for different MOPs (with three objectives), whereas Figures 8 to 18 show the parallel coordinates of Pareto fronts produced by the compared algorithms for different MOPs (with 10 objectives). Tables 3 and 5 show the mean (outside the parentheses) and the standard deviation (inside the parentheses) of the selected algorithms in terms of the mean IGD$^+$ value for different MOPs with different numbers of objectives. The best mean is shown in **boldface**. The Wilcoxon rank-sum test at a 0.05 significance level was conducted between the proposed algorithm and the nine popular multi-objective optimization algorithms that we selected, respectively. In Tables 3 and 5, †, ‡ and = are marked next to the values of an algorithm in the tables to denote that the performance of the algorithm is significantly better, worse or has no significant difference with respect to that of MOPSO-HGLSS.

WFG1 is separable and unimodal. Its Pareto optimal front is both concave and convex. In Table 3, it is shown that MOPSO-HGLSS always outperforms SMPSO, dMOPSO and MOPSOhv when WFG1 is scaled from three to ten objectives (3, 5, 8 and 10 objectives) in terms of the mean IGD$^+$ value. However, MaPSO and ARMOEA outperform the proposed algorithm. MOEA/D performs better than MOPSO-HGLSS in WFG1-3 but it is outperformed when the number of objectives is 5, 8 and 10. For DBEA, it outperforms the proposed algorithm when using 3 and 8 objectives, but is outperformed when the number of objectives is 10. For NSGA-III, it outperforms the proposed algorithm when the problem has 3 objectives, but it is outperformed when the number of objectives is 5, 8 and 10. For RVEA, it has similar performance with the proposed algorithm when using 3 objectives, while the proposed algorithm outperforms RVEA when the number of objectives is 5, 8 and 10.

The Pareto optimal front of WFG2 is disconnected and convex. Figure 2 shows the approximations of the true Pareto front of this problem produced by DBEA, NSGA-III, RVEA, ARMOEA and our proposed algorithm. Figure 9 shows how both SMPSO and our proposed algorithm can generate well-distributed Pareto fronts. Regarding the mean IGD$^+$

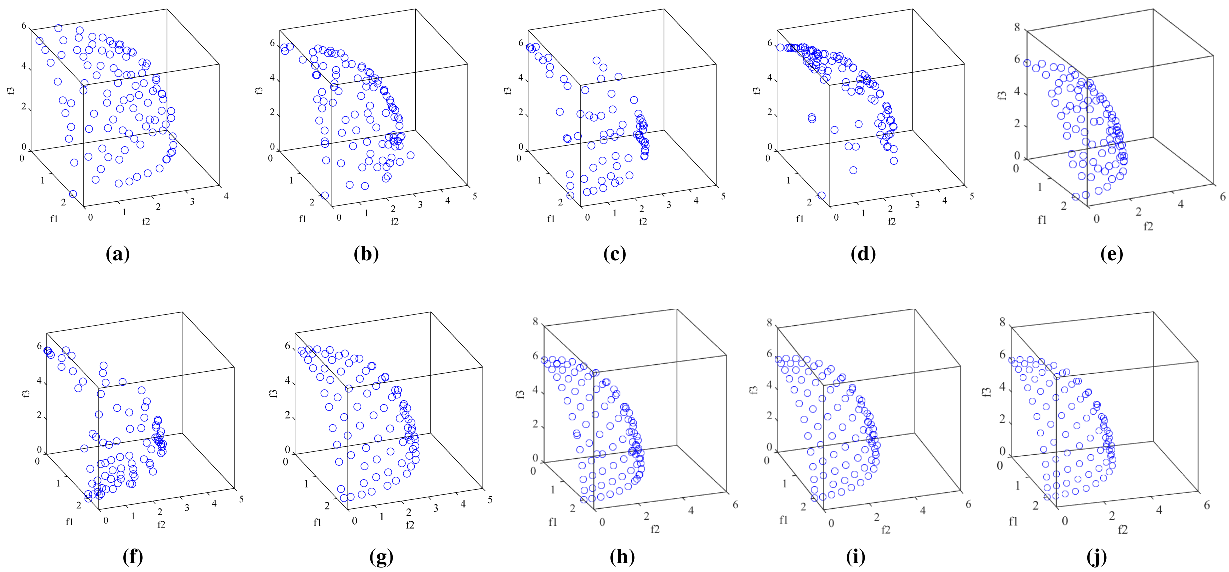


FIGURE 3. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG4-3, respectively.

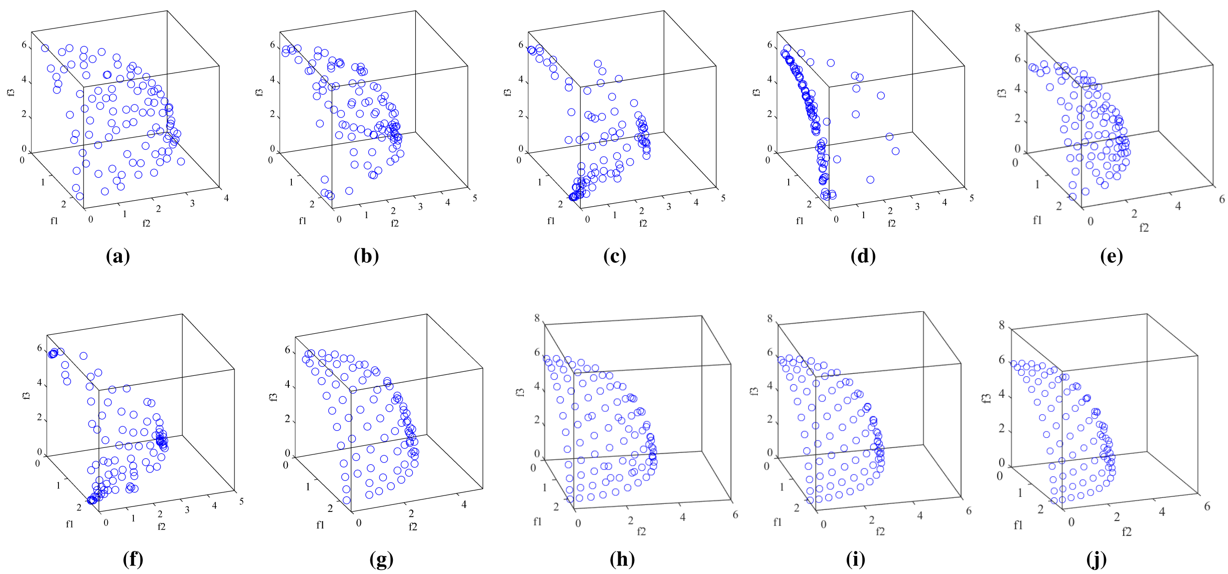


FIGURE 4. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG5-3, respectively.

value, our proposed algorithm performs better than the others when the problem is scaled up to 8 and 10 objectives.

The Pareto optimal front of WFG3 is degenerated and linear. Note that MOPSO-HGLSS performs better than SMPSO, MaPSO, DBEA and ARMOEA when WFG3 was scaled from 3 to 10 objectives. Although the performance of MOPSO-HGLSS is worse than that of dMOPSO, MOPSOhv, MOEA/D, NSGA-III and RVEA in some test instances, it outperforms them when the problem is scaled up to 8 and 10 objectives.

WFG4 is multimodal and its Pareto optimal front is concave. It should be noted that our proposed algorithm outperforms MaPSO and ARMOEA when the problem is scaled from three to ten objectives in terms of the mean IGD$^+$ value. DBEA and RVEA outperform the proposed algorithm when the problem has 3 and 5 objectives, but they are outperformed when the problem has 8 and 10 objectives. Besides, our proposed algorithm outperforms dMOPDO, MOPSOhv, MOEA/D and NSGA-III when the problem has 8 and 10 objectives. Figure 11 shows that the performance of our proposed algorithm improves when the problem is scaled up to 10 objectives.

WFG5 is deceptive and separable. Its Pareto optimal front is concave. Regarding IGD$^+$, our proposed algorithm performs better than SMPSO, dMOPSO, MOPSOhv, MaPSO and MOEA/D when the problem is scaled from 3 to

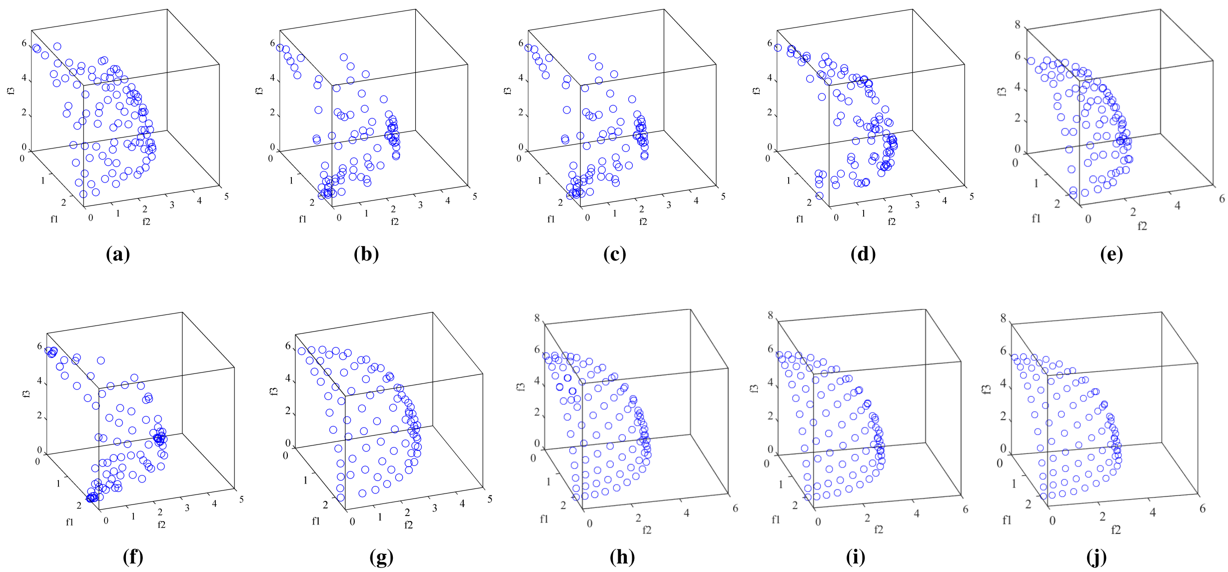


FIGURE 5. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG6-3, respectively.

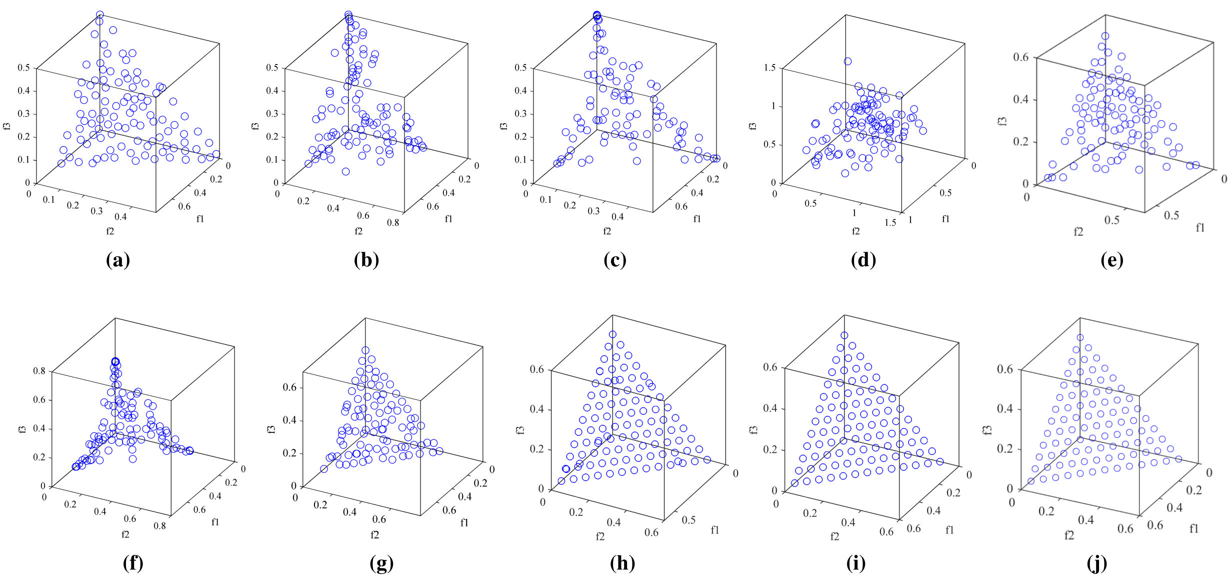


FIGURE 6. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem DTLZ1-3, respectively.

10 objectives in terms of mean IGD$^+$ value. It also outperforms DBEA and NSGA-III when the problem has 10 objectives. Note that RVEA has similar performance than our proposed algorithm when the problem has 3 and 10 objectives. Although Figure 4 shows that DBEA, NSGA-III, RVEA and ARMOEA produce better-distributed Pareto fronts than our proposed algorithm, Figure 12 shows that the performance of our proposed algorithm improves when the problem is scaled up to 10 objectives. Additionally, Table 3 shows that our proposed algorithm obtains the smallest IGD$^+$ value when the problem has 10 objectives.

WFG6 is non-separable and unimodal. Its Pareto optimal front is concave in shape. The performance of our proposed algorithm is not satisfactory when the problem has 3 and 5 objectives, respectively. However, as the problem is scaled up, MOPSO-HGLSS outperforms the other algorithms in terms of IGD$^+$. Similar to WFG5, Figure 5 shows that DBEA, NSGA-III, RVEA and ARMOEA can generate better-distributed Pareto fronts than our proposed algorithm. However, Figure 13 shows that the performance of our proposed algorithm improves when the problem is scaled up to 10 objectives. Additionally, Table 3 shows that our proposed

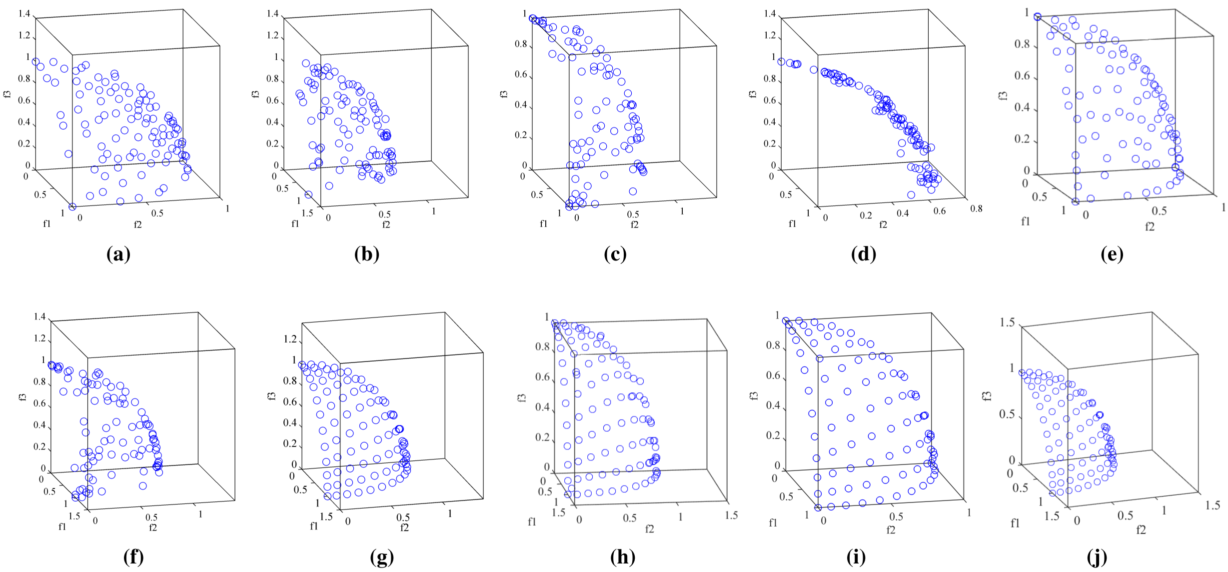


FIGURE 7. Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem DTLZ2-3, respectively.

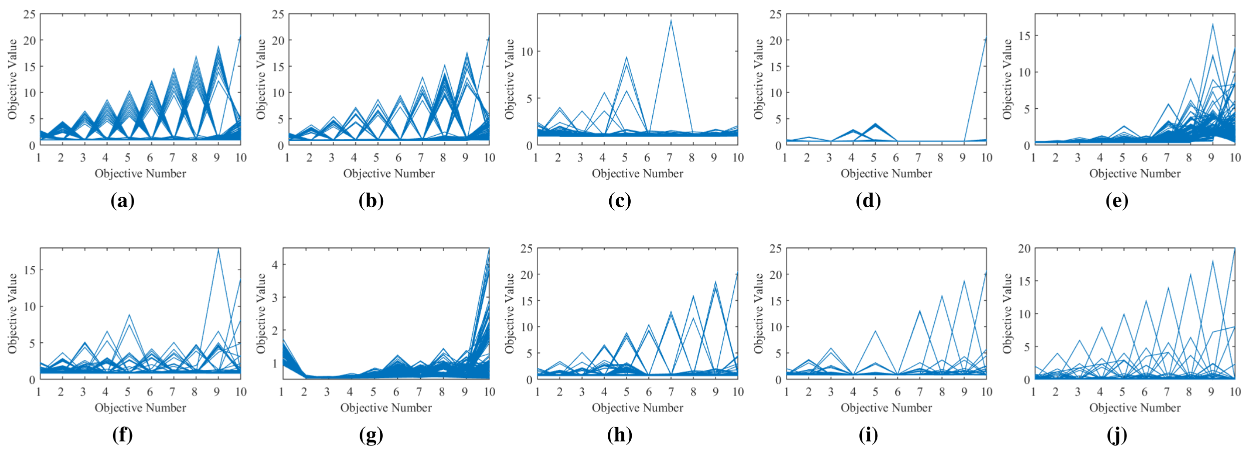


FIGURE 8. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG1-10, respectively.

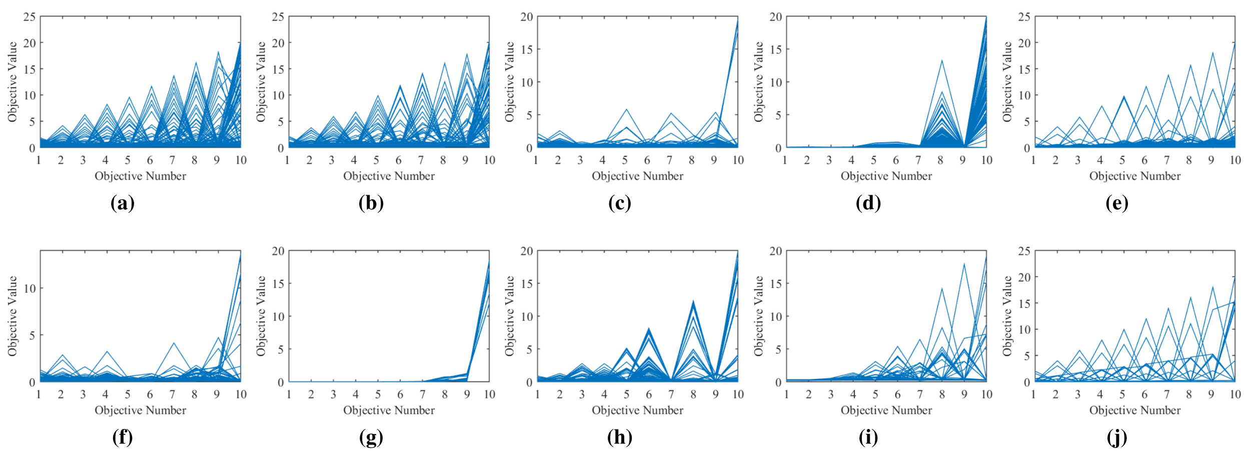


FIGURE 9. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG2-10, respectively.

algorithm obtains the smallest IGD$^+$ value when the problem has 10 objectives.

WFG7 is separable, unimodal and parameter dependent. Its Pareto optimal front is concave. Regarding IGD$^+$,

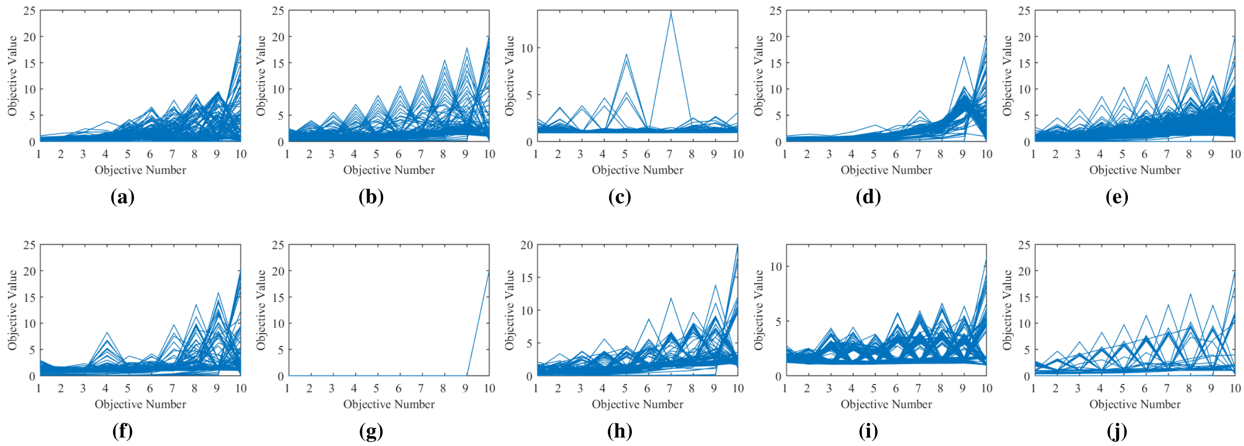


FIGURE 10. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG3-10, respectively.

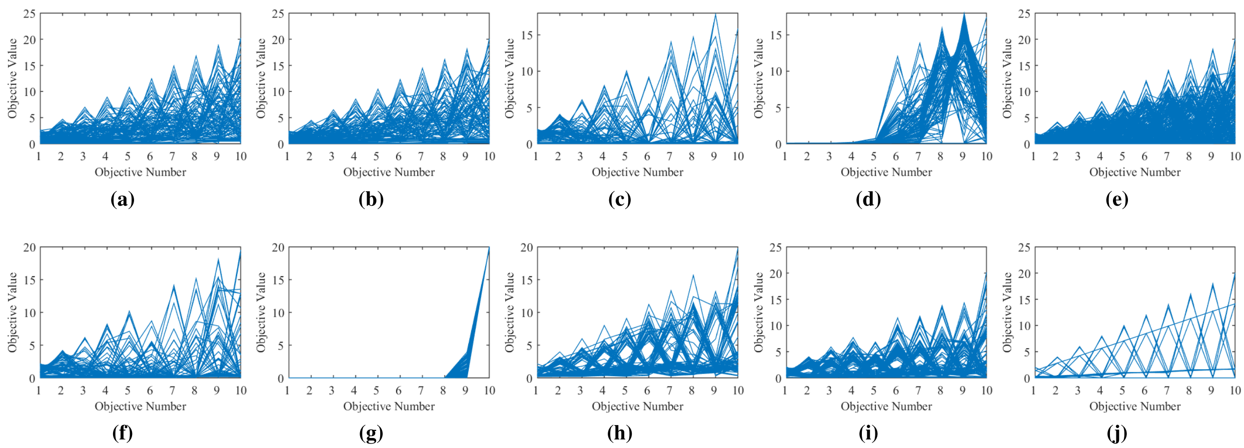


FIGURE 11. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG4-10, respectively.

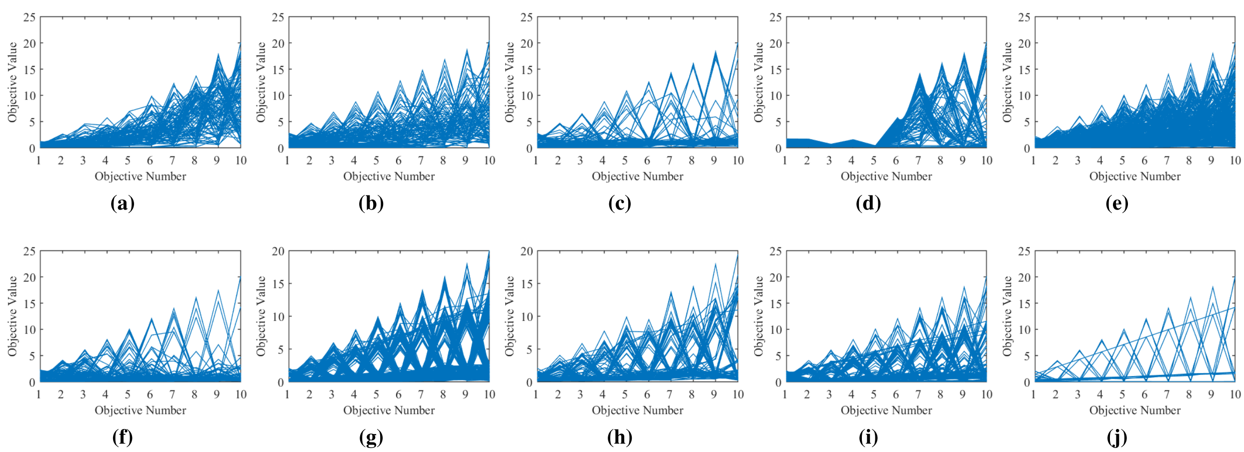


FIGURE 12. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG5-10, respectively.

our proposed approach outperforms the compared algorithm except for NSGA-III and RVEA. RVEA has a similar performance to that of our proposed algorithm when the problem has 3 and 5 objectives. Figure 14 shows that both SMPSO and our proposed algorithm can generate well-distributed Pareto fronts when the problem has 10 objectives.

WFG8 is unimodal and parameter dependent but non-separable. Its Pareto optimal front is concave. Although

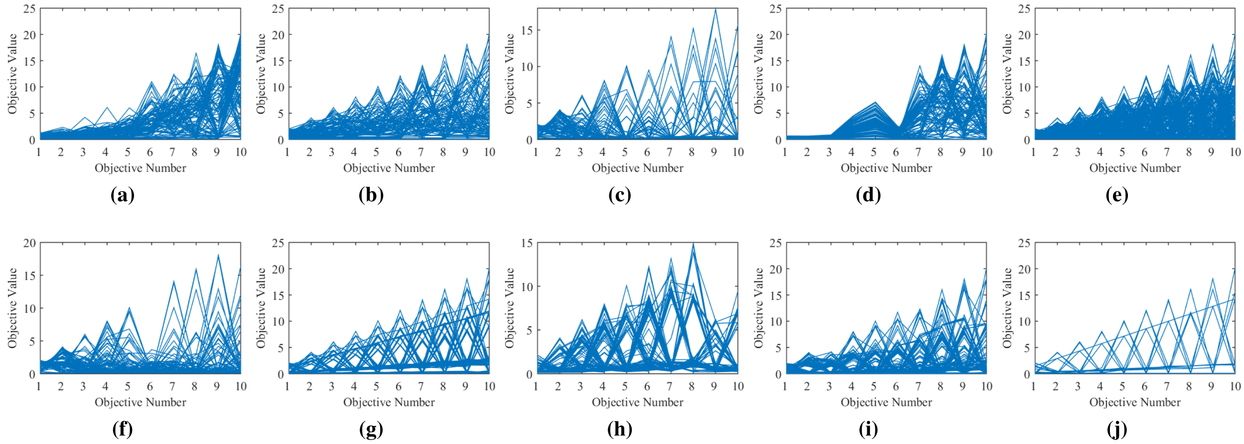


FIGURE 13. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG6-10, respectively.

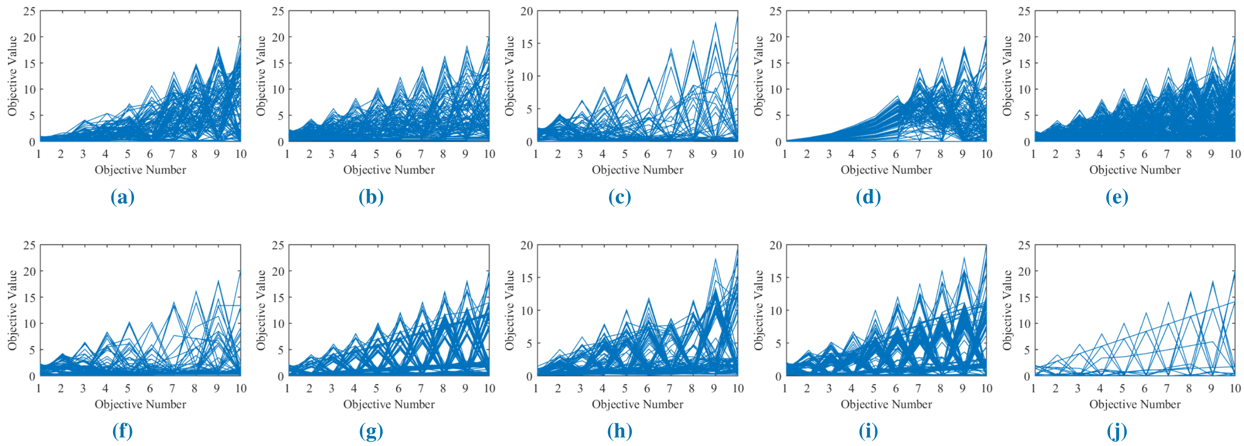


FIGURE 14. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG7-10, respectively.

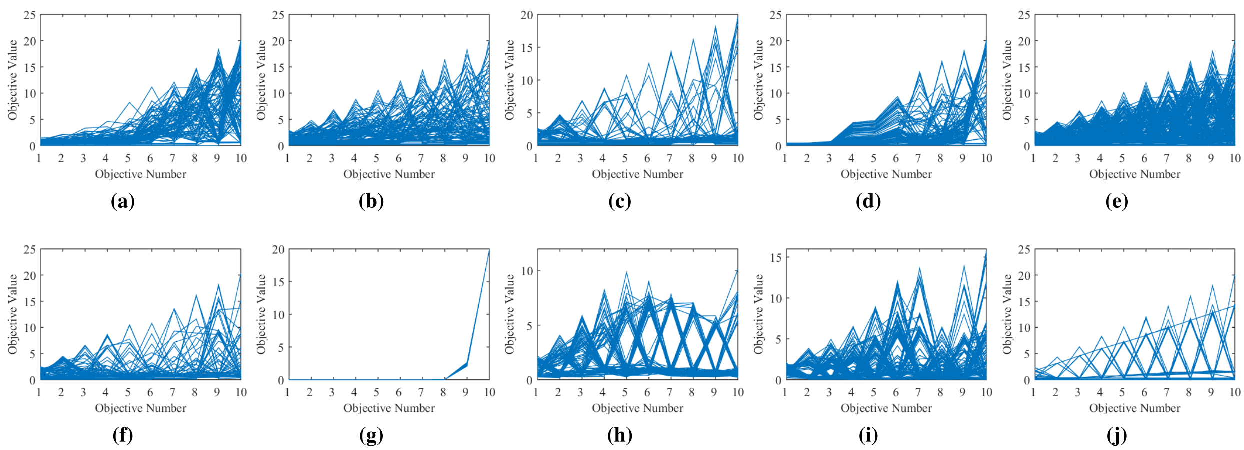


FIGURE 15. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG8-10, respectively.

the performance of our proposed algorithm is not satisfactory in WFG8-3, WFG8-5 and WFG8-8 in terms of IGD$^+$, its performance improves when the problem is scaled up to 10 objectives. As shown in Figure 15, our proposed algorithm is not the best optimizer whereas SMPSO can generate a well-distributed Pareto front when the problem has 10 objectives.

WFG9 is non-separable, multimodal, parameter dependent and deceptive. Its Pareto optimal front is concave. Table 3 shows that the scalability of our proposed approach is not as

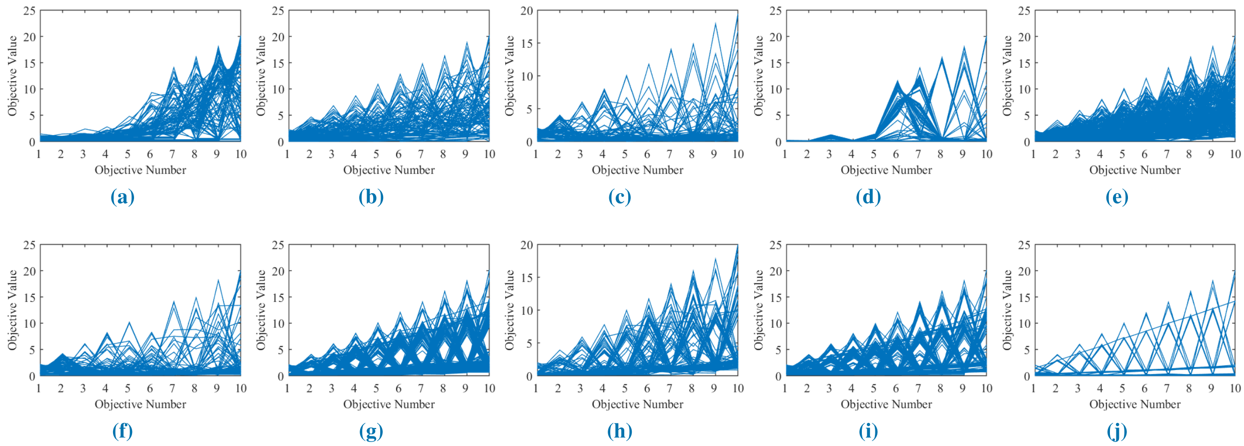


FIGURE 16. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem WFG9-10, respectively.

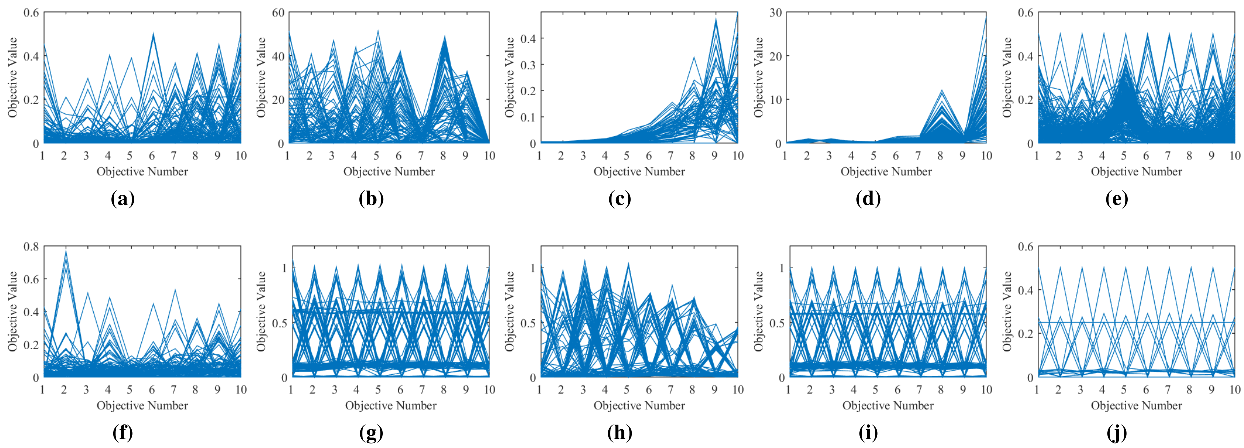


FIGURE 17. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem DTLZ1-10, respectively.

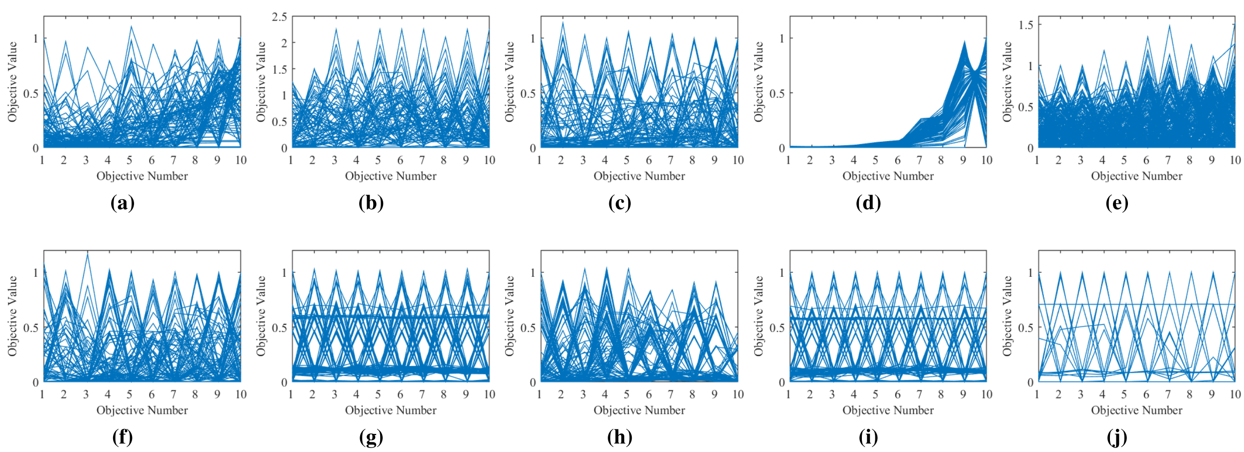


FIGURE 18. The parallel coordinates of Pareto fronts produced by (a) MOPSO-HGLSS, (b) SMPSO, (c) dMOPSO, (d) MOPSOhv, (e) MaPSO, (f) MOEA/D, (g) DBEA, (h) NSGA-III, (i) RVEA and (j) ARMOEA on problem DTLZ2-10, respectively.

good as that of the other algorithms under WFG9 in terms of mean IGD$^+$ value. Furthermore, Figure 16 shows that SMPSO can generate a well-distributed Pareto front when the problem has 10 objectives.

TABLE 3. Performance comparisons of different algorithms in terms of the mean IGD$^+$ value for DTLZ1, DTLZ2, and WFG1 to WFG9.

	M	MOPSO-HGLSS	SMPSO	dMOPSO	MOPSOhv	MaPSO	MOEA/D	DBEA	NSGA-III	RVEA	ARMOEA
WFG1	3	4.73E-1(3.8E-2)	6.06E-1(7.2E-3)†	5.29E-1(5.8E-3)†	6.81E-1(4.2E-2)†	2.21E-1(8.2E-2)‡	4.30E-1(1.3E-2)‡	3.91E-1(2.0E-2)‡	4.36E-1(2.6E-2)‡	4.68E-1(4.6E-2)=	**1.47E-1**(9.0E-4)‡
	5	5.52E-1(3.6E-2)	7.44E-1(6.0E-3)†	6.85E-1(8.1E-3)†	7.02E-1(1.8E-1)†	**3.55E-1**(8.8E-2)‡	5.90E-1(1.1E-2)†	5.43E-1(7.4E-2)=	7.68E-1(3.2E-2)†	6.10E-1(1.9E-2)†	4.74E-1(5.2E-3)‡
	8	1.31E+0(3.7E-2)	1.62E+0(1.0E-2)†	1.73E+0(3.8E-2)†	1.50E+0(5.6E-2)†	**7.03E-1**(2.6E-1)‡	1.43E+0(3.5E-2)†	9.96E-1(5.7E-2)‡	1.70E+0(5.9E-2)†	1.42E+0(6.0E-2)†	1.06E+0(2.9E-2)‡
	10	3.53E+0(5.9E-2)	4.20E+0(2.0E-2)†	4.25E+0(6.1E-2)†	3.68E+0(1.9E-1)†	**8.09E-1**(2.4E-1)‡	3.86E+0(2.3E-1)†	4.27E+0(2.4E+0)†	4.18E+0(1.7E-1)†	3.67E+0(1.8E-1)†	1.59E+0(4.7E-2)‡
WFG2	3	2.22E-2(3.3E-3)	5.11E-2(8.3E-3)†	2.69E-2(3.3E-3)†	3.29E-2(8.6E-3)†	1.28E-1(7.9E-3)†	**2.18E-2**(2.7E-3)=	3.45E-2(3.1E-3)†	2.87E-2(3.0E-3)†	3.37E-2(3.2E-3)†	1.65E-1(8.7E-4)†
	5	5.41E-2(5.9E-3)	1.71E-1(2.6E-2)†	5.50E-2(3.6E-3)=	**4.42E-2**(1.3E-2)‡	3.44E-1(1.1E-2)†	5.09E-2(1.2E-2)‡	1.12E-1(3.7E-2)†	9.75E-2(3.5E-2)†	5.50E-2(8.4E-3)=	5.04E-1(2.7E-3)†
	8	**7.78E-2**(8.2E-3)	2.04E-1(4.0E-2)†	1.14E-1(2.0E-2)†	8.39E-2(2.1E-2)†	4.01E-1(9.0E-3)†	1.20E-1(3.5E-2)†	2.99E-1(9.2E-2)†	1.48E-1(4.9E-2)†	1.74E-1(5.4E-2)†	1.09E+0(2.7E-2)†
	10	**1.36E-1**(2.9E-2)	5.16E-1(1.1E-1)†	1.75E-1(4.5E-2)†	1.49E-1(2.1E-2)†	4.94E-1(9.2E-3)†	3.46E-1(1.3E-1)†	3.41E-1(7.0E-2)†	1.77E-1(6.5E-2)†	6.65E-1(2.2E-1)†	1.37E+0(2.8E-2)†
WFG3	3	3.87E-2(4.8E-3)	7.39E-2(1.6E-2)†	3.37E-2(5.1E-3)‡	1.56E-2(8.8E-3)‡	1.01E-1(3.2E-3)†	**1.29E-2**(1.2E-3)‡	4.38E-2(7.0E-3)†	3.87E-2(6.0E-3)=	8.14E-2(9.0E-3)†	1.11E-1(8.0E-3)†
	5	2.28E-1(4.8E-2)	4.99E-1(3.7E-2)†	2.27E-1(2.5E-2)=	**2.41E-2**(5.0E-3)‡	4.62E-1(1.2E-2)†	1.06E-1(2.7E-2)‡	5.00E-1(2.6E-4)†	1.70E-1(2.7E-2)‡	1.89E-1(3.6E-2)‡	6.83E-1(4.9E-2)†
	8	**4.60E-1**(6.3E-2)	4.50E+0(6.2E-1)†	7.06E-1(2.0E-1)†	1.33E+0(4.0E-1)†	9.18E-1(1.4E-2)†	5.25E-1(9.5E-2)†	5.13E-1(1.7E-2)†	2.19E+0(8.2E-1)†	1.75E+1(6.1E+0)†	2.27E+0(1.7E-1)†
	10	**5.51E-1**(8.2E-2)	1.80E+1(3.2E+0)†	2.66E+0(1.4E+0)†	7.08E+0(1.8E+0)†	1.17E+0(1.0E-2)†	8.77E+0(3.9E+0)†	7.38E-1(1.7E-1)†	8.75E+0(2.9E+0)†	1.36E+2(3.9E+1)†	3.32E+0(1.6E-1)†
WFG4	3	3.01E-2(5.8E-3)	5.52E-2(1.9E-3)†	4.73E-2(1.3E-3)†	4.97E-2(1.0E-2)†	2.02E-1(2.8E-3)†	4.44E-2(4.0E-3)†	2.78E-2(1.2E-3)‡	**2.10E-2**(2.8E-4)‡	2.36E-2(8.6E-4)‡	2.21E-1(3.5E-5))†
	5	9.98E-2(8.3E-3)	9.31E-2(6.5E-3)‡	1.05E-1(5.0E-3)=	3.51E-1(2.8E-1)†	1.03E+0(1.9E-2)†	8.69E-2(9.8E-3)=	**4.90E-2**(1.3E-2)‡	1.55E-1(3.7E-2)†	5.98E-2(7.5E-3)‡	1.23E+0(6.4E-4)†
	8	**1.26E-1**(1.0E-2)	1.29E-1(9.0E-3)=	4.26E-1(1.4E-1)†	3.79E-1(2.5E-1)†	2.49E+0(3.3E-2)†	2.04E-1(9.7E-2)†	1.52E-1(7.2E-2)†	2.76E-1(1.0E-1)†	4.15E-1(1.5E-1)†	3.54E+0(7.2E-3)†
	10	**1.51E-1**(1.3E-2)	1.56E-1(1.4E-2)=	5.75E-1(1.4E-1)†	4.36E-1(2.2E-1)†	3.42E+0(3.3E-2)†	1.86E-1(5.2E-2)†	1.80E-1(6.7E-2)†	5.27E-1(1.1E-1)†	6.88E-1(3.0E-1)†	5.88E+0(1.3E-2)†
WFG5	3	4.38E-2(1.8E-3)	6.01E-2(3.9E-3)†	5.70E-2(2.4E-4)†	8.44E-2(1.6E-2)†	1.98E-1(7.6E-3)†	5.81E-2(2.2E-3)†	4.29E-2(1.1E-3)=	**4.16E-2**(1.7E-4)‡	4.43E-2(1.9E-4)=	4.20E-2(1.1E-5)‡
	5	1.11E-1(4.9E-3)	1.46E-1(5.5E-3)†	1.23E-1(3.7E-3)†	2.05E-1(1.1E-1)†	9.96E-1(2.2E-2)†	1.17E-1(6.4E-3)†	3.86E-1(1.1E-3)†	1.28E-1(1.6E-2)†	**6.97E-2**(6.8E-4)‡	1.22E+0(1.3E-4)†
	8	1.53E-1(6.9E-3)	1.73E-1(1.2E-2)†	2.08E-1(4.2E-2)†	2.58E-1(1.4E-1)†	2.28E+0(2.7E-2)†	2.69E-1(9.7E-2)†	**1.05E-1**(5.2E-3)‡	1.88E-1(3.9E-2)†	1.23E-1(1.8E-3)‡	3.53E+0(7.4E-3)†
	10	**1.91E-1**(9.1E-3)	2.02E-1(8.7E-3)†	2.34E-1(6.1E-2)†	2.58E-1(1.1E-1)†	3.12E+0(1.6E-2)†	3.05E-1(1.1E-1)†	3.87E-1(1.7E-8)†	3.73E-1(9.5E-2)†	1.99E-1(8.5E-3)=	5.81E+0(2.5E-2)†
WFG6	3	5.72E-2(2.1E-3)	5.28E-2(4.4E-3)‡	4.11E-2(1.0E-3)‡	4.00E-2(3.7E-3)‡	6.96E-2(5.9E-3)†	5.23E-2(5.7E-3)‡	4.32E-2(1.8E-3)‡	**3.60E-2**(2.3E-3)‡	3.87E-2(3.7E-3)‡	4.38E-2(3.5E-3)‡
	5	1.20E-1(1.1E-2)	1.31E-1(6.0E-3)†	1.22E-1(5.8E-3)=	2.21E-1(1.2E-1)†	1.05E+0(2.1E-2)†	1.15E-1(9.8E-3)=	1.09E-1(2.6E-2)‡	1.42E-1(1.9E-2)†	**6.59E-2**(2.6E-3)‡	1.21E+0(7.6E-4)†
	8	**1.52E-1**(9.6E-3)	1.60E-1(6.6E-3)=	2.37E-1(7.8E-2)†	2.51E-1(1.3E-1)†	2.39E+0(3.3E-2)†	1.78E-1(6.0E-2)=	1.72E-1(1.1E-1)=	2.48E-1(1.1E-1)†	1.61E-1(5.2E-2)=	3.53E+0(9.9E-3)†
	10	**1.66E-1**(9.8E-3)	1.73E-1(6.5E-3)†	2.53E-1(1.0E-1)†	2.85E-1(1.4E-1)†	3.36E+0(3.3E-2)†	3.27E-1(1.3E-1)†	3.49E-1(1.0E-1)†	6.18E-1(9.5E-2)†	2.07E-1(1.0E-1)†	5.80E+0(3.3E-2)†
WFG7	3	**2.88E-2**(8.7E-4)	7.41E-2(5.4E-3)†	4.94E-2(1.2E-3)†	4.05E-2(3.3E-3)†	1.92E-1(5.2E-3)†	4.10E-2(1.6E-3)†	3.18E-2(1.0E-3)†	2.91E-2(8.6E-4)=	2.99E-2(8.4E-4)=	2.21E-1(3.9E-5)†
	5	**9.92E-2**(8.3E-3)	1.67E-1(5.2E-3)†	1.45E-1(4.7E-3)†	2.40E-1(1.3E-1)†	1.01E+0(1.9E-2)†	1.16E-1(7.1E-3)†	1.36E-1(2.7E-2)†	1.40E-1(2.4E-2)†	9.95E-2(6.4E-3)=	1.23E+0(1.3E-3)†
	8	**1.66E-1**(9.8E-3)	2.02E-1(5.4E-3)†	2.50E-1(4.0E-2)†	2.48E-1(1.1E-1)†	2.41E+0(3.4E-2)†	1.89E-1(3.4E-2)†	3.15E-1(9.7E-2)†	1.83E-1(1.9E-2)†	1.71E-1(1.9E-2)†	3.55E+0(1.7E-2)†
	10	**1.78E-1**(9.3E-3)	2.17E-1(8.2E-3)†	2.52E-1(3.3E-2)†	2.16E-1(7.1E-2)†	3.35E+0(2.4E-2)†	2.91E-1(8.5E-2)†	3.40E-1(8.4E-2)†	3.06E-1(4.9E-2)†	1.89E-1(2.4E-2)†	5.91E+0(9.4E-2)†
WFG8	3	6.48E-2(4.1E-3)	9.98E-2(5.7E-3)†	7.96E-2(3.0E-3)†	7.53E-2(4.5E-3)†	1.94E-1(4.6E-3)†	5.96E-2(2.5E-3)‡	6.26E-2(1.4E-3)=	**5.55E-2**(2.0E-3)‡	6.42E-2(4.6E-3)=	2.69E-1(2.8E-3)†
	5	1.18E-1(9.8E-3)	1.67E-1(8.0E-3)†	1.60E-1(6.0E-3)†	1.31E-1(1.4E-2)†	1.05E+0(2.4E-2)†	1.08E-1(7.5E-3)‡	2.86E-1(2.0E-1)†	2.03E-1(1.7E-2)†	**9.79E-2**(1.7E-2)‡	1.22E+0(8.0E-4)†
	8	1.71E-1(7.6E-3)	1.88E-1(1.0E-2)†	2.08E-1(2.5E-2)†	2.01E-1(1.1E-1)†	2.56E+0(3.3E-2)†	**1.28E-1**(1.2E-2)‡	3.98E-1(6.6E-2)†	4.27E-1(1.4E-1)†	6.08E-1(1.8E-1)†	3.61E+0(2.2E-2)†
	10	**2.02E-1**(9.2E-3)	2.06E-1(1.4E-2)=	2.05E-1(2.1E-2)=	2.03E-1(7.6E-2)=	3.52E+0(2.1E-2)†	2.14E-1(2.3E-2)=	3.77E-1(4.2E-2)†	8.34E-1(3.1E-2)†	9.19E-1(1.8E-1)†	5.98E+0(1.4E-1)†
WFG9	3	4.39E-2(3.1E-3)	3.59E-2(4.4E-3)‡	4.28E-2(7.3E-4)‡	7.16E-2(2.0E-2)†	2.03E-1(9.4E-3)†	5.78E-2(1.0E-2)†	**2.95E-2**(9.6E-3)‡	3.58E-2(9.9E-3)‡	3.18E-2(9.6E-3)‡	2.21E-1(6.6E-4)†
	5	1.08E-1(9.8E-3)	1.05E-1(1.3E-2)=	1.14E-1(7.7E-3)†	2.69E-1(2.4E-1)†	9.76E-1(2.0E-2)†	1.07E-1(1.8E-2)=	2.19E-1(7.9E-3)†	1.62E-1(3.3E-2)†	**1.07E-1**(1.3E-2)=	1.21E+0(3.9E-3)†
	8	5.46E-1(2.1E-2)	**3.29E-1**(9.8E-2)‡	4.11E-1(1.4E-1)‡	5.23E-1(1.9E-1)=	2.31E+0(2.7E-2)†	4.32E-1(1.8E-1)‡	5.29E-1(1.3E-1)=	7.84E-1(6.7E-2)†	8.97E-1(1.4E-1)†	3.53E+0(2.4E-2)†
	10	1.38E+0(1.5E-2)	**4.24E-1**(1.4E-1)‡	5.45E-1(2.3E-1)‡	1.06E+0(3.6E-1)‡	3.16E+0(2.5E-2)†	1.50E+0(2.3E-2)†	1.19E+0(5.4E-1)‡	1.96E+0(2.5E-1)†	2.71E+0(7.2E-1)†	5.78E+0(2.4E-2)†
DTLZ1	3	3.29E-2(7.9E-4)	5.19E-2(2.3E-3)†	3.81E-2(8.2E-4)†	1.38E+0(1.2E-2)†	**2.01E-2**(4.8E-4)‡	3.41E-2(6.1E-4)†	3.93E-2(2.9E-2)†	3.52E-2(1.1E-2)†	3.34E-2(2.5E-4)=	2.06E-2(1.1E-5)‡
	5	9.13E-2(3.4E-3)	2.44E-1(5.3E-2)†	1.34E-1(7.1E-3)†	1.58E+0(1.0E-1)†	7.51E-2(5.2E-3)‡	9.89E-2(2.7E-3)†	6.19E-1(4.1E-1)†	9.73E-2(2.6E-3)†	9.15E-2(2.4E-3)=	**6.82E-2**(8.1E-5)‡
	8	1.54E-1(6.5E-3)	3.93E+0(8.7E+0)†	2.08E-1(1.1E-2)†	1.73E+0(1.4E-2)†	1.26E-1(6.5E-3)‡	1.60E-1(7.5E-3)†	3.58E+0(3.1E+0)†	1.64E-1(7.3E-3)†	1.56E-1(6.5E-3)=	**1.08E-1**(1.1E-3)‡
	10	1.86E-1(1.2E-2)	8.11E+0(1.5E+1)†	2.60E-1(1.5E-2)†	3.32E+0(1.4E+0)†	**1.45E-1**(5.9E-3)‡	1.92E-1(1.2E-2)†	3.28E+0(2.5E+0)†	1.84E-1(1.1E-2) =	1.81E-1(1.2E-2)=	1.55E-1(6.8Ee-3)‡
DTLZ2	3	2.81E-2(3.1E-3)	4.24E-2(2.1E-3)†	3.58E-2(5.7E-4)†	2.20E-1(3.9E-2)†	4.94E-2(1.7E-3)†	3.42E-2(1.1E-3)†	2.38E-2(9.7E-4)‡	**2.25E-2**(2.8E-4)‡	2.26E-2(6.2E-6)‡	5.45E-2(4.4E-6)†
	5	1.35E-1(9.1E-3)	3.32E-1(3.2E-2)†	1.15E-1(2.1E-3)‡	4.00E-1(2.0E-2)†	1.86E-1(4.2E-3)†	1.06E-1(3.6E-3)‡	4.52E-1(4.2E-2)†	1.16E-1(5.5E-3)‡	**6.34E-2**(3.2E-5)‡	2.12E-1(3.3E-5)†
	8	2.51E-1(2.1E-2)	7.68E-1(9.7E-2)†	2.01E-1(3.0E-3)‡	5.52E-1(1.7E-2)†	3.20E-1(5.5E-3)†	1.86E-1(4.1E-3)‡	2.38E-1(1.6E-1)=	1.92E-1(1.3E-2)‡	**1.41E-1**(1.2E-4)‡	3.87E-1(6.8E-4)†
	10	3.49E-1(1.9E-2)	1.06E+0(1.5E-1)†	2.47E-1(8.1E-3)‡	6.29E-1(1.9E-2)†	3.83E-1(3.8E-3)†	2.73E-1(1.9E-2)‡	7.97E-1(5.6E-7)†	2.98E-1(1.7E-2)‡	**1.46E-1**(2.4E-4)‡	5.02E-1(4.7E-3)†

TABLE 4. Summary of Wilcoxon rank-sum test results for the selected algorithms with respect to the mean IGD$^+$ value for DTLZ1, DTLZ2, and WFG1 to WFG9.

	SMPSO	dMOPSO	MOPSOhv	MaPSO	MOEA/D	DBEA	NSGA-III	RVEA	ARMOEA
†	34	31	37	36	26	28	30	18	34
‡	5	8	5	8	12	10	11	13	10
=	5	5	2	0	6	6	3	13	0

†, ‡ and = denote the number of times the performance of the corresponding algorithm is significantly better, worse or has no significant difference with respect to that of the proposed algorithm, respectively.

DTLZ1 is multimodal and its Pareto optimal front is linear. Regarding the mean IGD$^+$ value, our proposed approach performs better than SMPSO, dMOPSO, MOPSOhv, MOEA/D and DBEA when the problem is scaled from 3 to 10 objectives. For NSGA-III, its performance is worse than that of our proposed algorithm when the problem has 3, 5 and 8 objectives. For RVEA, it has a similar performance as our proposed algorithm when the problem has 3, 5, 8 and 10 objectives. However, MaPSO and ARMOEA outperform our proposed algorithm. Both Figures 6 and 17 show that RVEA and ARMOEA work perform in this problem.

DTLZ2 is unimodal and its Pareto optimal front is concave. Note that MOPSO-HGLSS performs better than SMPSO and MOPSOhv when the problem is scaled from 3 to 10 objectives in terms of mean IGD$^+$ value. However, dMOPSO and MOEA/D perform better than MOPSO-HGLSS for DTLZ2-5, DTLZ2-8 and DTLZ2-10. For MaPSO and ARMOEA, they are outperformed by our proposed algorithm in terms of mean IGD$^+$ values. Regarding DBEA, it outperforms our proposed algorithm for DTLZ2-3 but it is outperformed when the number of objectives is 5 and 10 in terms of mean IGD$^+$ value. For NSGA-III and RVEA, they outperform our proposed algorithm when the problem

TABLE 5. Performance comparisons of different algorithms in terms of the mean IGD$^+$ value for MaF1 to MaF8.

	M	MOPSO-HGLSS	SMPSO	dMOPSO	MOPSOhv	MaPSO	MOEA/D	DBEA	NSGA-III	RVEA	ARMOEA
MaF1	3	4.55E-2(1.1E-3)	6.10E-2(3.0E-3)†	1.09E-1(5.4E-3)†	6.54E-2(2.2E-3)†	**3.53E-2**(1.4E-3)‡	7.04E-2(4.1E-7)†	6.99E-2(2.8E-4)†	6.28E-2(1.7E-3)†	8.19E-2(7.6E-4)†	4.34E-2(1.6E-4)‡
	5	1.58E-1(1.0E-2)	1.89E-1(8.8E-3)†	3.42E-1(4.3E-2)†	1.94E-1(7.4E-3)†	**1.12E-1**(2.6E-3)‡	2.25E-1(1.6E-3)†	2.28E-1(3.5E-6)†	2.53E-1(3.9E-2)†	3.54E-1(6.7E-2)†	1.59E-1(2.8E-3)=
	8	2.86E-1(2.1E-2)	3.18E-1(1.1E-2)†	5.30E-1(4.4E-2)†	3.19E-1(1.2E-2)†	**1.86E-1**(1.2E-3)‡	5.14E-1(1.3E-3)†	3.52E-1(4.1E-3)†	2.85E-1(9.0E-3)=	7.13E-1(5.5E-2)†	2.87E-1(1.4E-3)=
	10	3.37E-1(2.3E-2)	3.63E-1(1.4E-2)†	5.73E-1(2.8E-2)†	3.94E-1(2.7E-2)†	**2.22E-1**(4.2E-3)‡	5.34E-1(3.0E-4)†	4.01E-1(1.0E-2)†	3.41E-1(9.0E-3)=	6.69E-1(8.2E-2)†	3.11E-1(6.8E-4)‡
MaF2	3	5.14E-2(4.1E-3)	6.26E-2(3.6E-3)†	4.35E-2(8.7E-4)‡	6.07E-2(3.9E-3)†	**2.45E-2**(8.8E-4)‡	3.84E-2(3.5E-4)‡	4.66E-2(1.1E-3)‡	3.65E-2(5.6E-4)‡	4.30E-2(1.5E-3)‡	3.31E-2(8.6E-4)‡
	5	1.72E-1(4.7E-3)	1.48E-1(7.8E-3)‡	1.28E-1(2.2E-3)‡	1.53E-1(6.6E-3)‡	**8.09E-2**(7.2E-4)‡	1.40E-1(1.2E-3)‡	1.52E-1(6.1E-4)‡	1.42E-1(3.3E-3)‡	1.45E-1(1.3E-3)‡	1.22E-1(1.6E-3)‡
	8	2.33E-1(5.7E-3)	2.01E-1(4.1E-3)‡	2.46E-1(9.8E-3)†	1.99E-1(4.0E-3)‡	**1.22E-1**(4.1E-3)‡	2.32E-1(4.2E-4)=	1.98E-1(9.1E-4)‡	2.59E-1(6.3E-2)†	5.87E-1(2.1E-1)†	1.98E-1(3.9E-3)‡
	10	2.43E-1(5.2E-3)	1.97E-1(4.3E-3)‡	4.45E-1(2.8E-2)†	2.10E-1(2.3E-3)‡	**1.31E-1**(1.9E-3)‡	3.70E-1(3.7E-3)†	8.61E-1(5.7E-3)†	3.16E-1(4.7E-2)†	6.43E-1(1.3E-1)†	2.48E-1(1.2E-2)=
MaF3	3	2.99E-2(1.7E-3)	4.20E+3(5.1E+2)†	5.98E-1(1.7E-1)†	1.88E+4(6.7E+3)†	**2.74E-2**(1.7E-3)‡	5.41E-2(5.5E-4)†	5.00E-2(2.6E-3)†	4.65E-2(2.7E-4)†	4.06E-2(7.2E-4)†	4.66E-2(2.6E-4)†
	5	**5.15E-2**(5.5E-3)	2.86E+3(1.4E+3)†	5.41E-1(2.6E-2)†	3.33E+4(5.9E+3)†	1.21E-1(3.6E-2)†	1.24E-1(1.6E-3)†	1.37E+6(4.3E+6)†	9.88E-2(1.3E-3)†	7.97E-2(4.9E-3)†	9.88E-2(1.0E-3)†
	8	1.51E-1(3.3E-3)	7.12E+4(1.2E+4)†	6.54E-1(1.9E-1)†	5.51E+4(3.7E+4)†	3.28E+0(7.0E-1)†	1.65E-1(1.7E-3)†	4.16E+0(8.9E+0)†	1.48E+0(2.7E+0)†	**1.13E-1**(7.0E-3)‡	1.35E-1(3.8E-3)‡
	10	2.64E-1(1.1E-3)	5.67E+7(9.4E+6)†	5.39E-1(1.8E-2)†	1.34E+5(1.3E+5)†	4.23E+2(5.4E+1)†	1.90E-1(5.2E-4)‡	3.01E+2(5.8E+2)†	7.09E+1(1.6E+2)†	**9.59E-2**(4.2E-3)‡	1.14E-1(6.6E-3)‡
MaF4	3	**3.93E-2**(2.8E-3)	2.76E+1(3.8E+1)†	3.93E+1(7.4E+1)†	3.97E+2(8.6E+1)†	2.27E-1(3.9E-3)†	6.66E-1(1.7E-2)†	8.66E-1(4.1E-1)†	3.52E-1(2.0E-2)†	3.95E-1(7.4E-2)†	3.40E-1(1.6E-3)†
	5	**1.81E-1**(7.4E-3)	8.20E+1(1.4E+2)†	1.14E+3(1.1E+3)†	2.51E+3(2.8E+2)†	1.93E+0(5.1E-2)†	9.92E+0(2.9E-1)†	4.97E+0(2.8E+0)†	3.91E+0(6.8E-1)†	4.46E+0(7.4E-1)†	2.89E+0(1.1E-1)†
	8	**6.59E-1**(2.7E-1)	1.08E+3(1.7E+3)†	1.70E+4(7.2E+3)†	1.86E+4(3.5E+3)†	1.63E+1(1.5E+0)†	1.18E+2(5.3E+0)†	3.80E+1(6.6E-1)†	3.45E+1(2.2E+0)†	7.50E+1(1.6E+1)†	3.17E+1(3.2E+0)†
	10	**2.37E+0**(1.5E+0)	6.89E+2(6.6E+2)†	5.42E+4(3.8E+4)†	6.46E+4(1.9E+4)†	6.59E+1(6.5E+0)†	4.44E+2(9.6E+0)†	1.41E+2(7.9E-1)†	1.55E+2(1.6E+1)†	2.12E+2(5.1E+1)†	1.35E+2(1.3E+1)†
MaF5	3	**1.04E-1**(2.3E-2)	9.91E-1(1.3E+0)†	7.17E-1(1.7E-1)†	4.11E-1(4.0E-2)†	1.97E-1(6.9E-3)†	5.38E-1(5.0E-1)†	8.97E-1(8.5E-1)†	4.37E-1(5.6E-1)†	2.59E-1(6.4E-6)†	1.67E+0(1.3E+0)†
	5	**2.51E-1**(2.1E-2)	3.56E+0(4.2E-1)†	1.13E+1(8.3E-2)†	4.07E+0(3.2E-1)†	1.47E+0(4.7E-2)†	9.39E+0(1.0E+0)†	6.00E+0(2.0E+0)†	2.73E+0(1.1E+0)†	2.54E+0(4.1E-1)†	2.40E+0(8.8E-2)†
	8	**1.03E+0**(8.5E-1)	3.88E+1(6.4E+0)†	8.73E+1(4.8E-2)†	3.38E+1(4.5E+0)†	1.00E+1(5.3E-1)†	8.45E+1(6.4E-1)†	3.42E+1(7.8E+0)†	2.82E+1(3.3E-2)†	3.09E+1(4.3E+0)†	2.87E+1(5.5E-1)†
	10	**6.41E+0**(1.9E+0)	1.30E+2(1.7E+1)†	3.06E+2(6.8E-1)†	1.17E+2(9.5E+0)†	3.28E+1(1.6E+0)†	3.03E+2(4.7E-1)†	1.40E+2(6.0E+1)†	1.37E+2(5.3E-1)†	1.29E+2(1.6E+1)†	1.62E+2(7.3E+0)†
MaF6	3	1.62E-1(6.1E-5)	5.31E-3(2.1E-4)‡	4.54E-1(8.3E-2)†	**4.65E-3**(1.8E-4)‡	2.29E-2(4.4E-3)‡	3.39E-2(3.3E-7)‡	1.82E-2(1.8E-3)‡	1.69E-2(2.2E-3)‡	3.65E-2(4.7E-3)‡	5.10E-3(1.1E-4)‡
	5	3.34E-1(5.7E-5)	5.70E-3(2.6E-4)‡	4.83E-1(1.4E-1)†	**5.06E-3**(2.1E-4)‡	5.12E-2(1.2E-2)‡	1.13E-1(1.9E-1)‡	1.14E-1(4.2E-3)‡	6.33E-2(1.6E-2)‡	8.18E-2(6.1E-3)‡	5.11E-3(6.3E-5)‡
	8	3.71E-1(4.4E-5)	7.28E-2(8.3E-2)‡	5.00E-1(1.4E-1)†	9.40E-1(3.4E-1)†	5.49E-2(2.4E-2)‡	6.35E-2(5.3E-2)‡	7.42E-1(1.9E-7)†	1.03E-1(6.3E-2)‡	3.90E-1(3.0E-1)‡	**6.34E-3**(4.2E-4)‡
	10	3.77E-1(3.4E-3)	4.07E-1(2.5E-1)†	5.15E-1(6.6E-2)†	1.59E+0(7.1E-1)†	1.50E+0(7.4E-1)†	4.50E-1(3.1E-1)†	7.42E-1(1.3E-7)†	3.25E-1(1.1E-2)‡	1.65E-1(1.0E-2)‡	**1.39E-2**(9.9E-3)‡
MaF7	3	**2.46E-2**(9.4E-4)	9.59E-2(1.1E-2)†	1.38E-1(5.3E-3)†	8.71E-2(3.6E-3)†	5.37E-2(1.4E-3)†	2.18E-1(2.0E-1)†	9.61E-2(4.2E-3)†	7.75E-2(4.3E-3)†	1.04E-1(1.0E-3)†	1.79E-1(1.4E-1)†
	5	**1.33E-1**(6.0E-3)	5.04E-1(9.2E-3)†	6.74E-1(1.4E-1)†	4.17E-1(9.3E-3)†	2.34E-1(5.0E-3)†	1.00E+0(1.6E-1)†	4.23E-1(2.6E-2)†	3.78E-1(1.0E-2)†	5.03E-1(7.5E-3)†	3.49E-1(8.6E-3)†
	8	**2.68E-1**(9.4E-3)	1.55E+0(5.7E-1)†	2.45E+0(7.7E-1)†	8.73E-1(9.0E-3)†	4.73E-1(2.8E-2)†	1.91E+0(1.8E-1)†	1.69E+0(9.8E-1)†	9.34E-1(7.3E-2)†	1.93E+0(6.6E-2)†	1.78E+0(1.1E-1)†
	10	**3.23E-1**(8.1E-3)	1.93E+0(1.3E-1)†	3.20E+0(1.0E+0)†	1.15E+0(2.1E-2)†	5.20E-1(2.3E-2)†	1.90E+0(3.0E-1)†	6.68E+0(4.8E+0)†	1.72E+0(1.8E-1)†	3.47E+0(5.5E-1)†	3.46E+0(1.9E-1)†
MaF8	3	**2.82E-2**(9.8E-4)	7.89E-2(3.7E-3)†	1.12E-1(8.6E-4)†	8.04E-2(2.5E-3)†	5.96E-2(3.7E-3)†	1.10E-1(2.7E-3)†	1.41E-1(1.1E-3)†	1.09E-1(4.3E-3)†	1.38E-1(8.6E-3)†	7.74E-2(2.1E-3)†
	5	**4.39E-2**(1.1E-3)	1.53E-1(8.7E-3)†	2.72E-1(6.0E-3)†	1.48E-1(4.8E-3)†	1.01E-1(2.8E-3)†	2.86E-1(7.5E-3)†	2.07E-1(8.3E-3)†	2.43E-1(2.1E-2)†	4.72E-1(4.2E-2)†	1.38E-1(5.2E-3)†
	8	**5.40E-2**(4.3E-4)	2.05E-1(5.4E-3)†	7.01E-1(2.9E-3)†	2.08E-1(7.4E-3)†	1.29E-1(2.7E-3)†	7.67E-1(1.6E-2)†	6.48E-1(2.5E-2)†	4.39E-1(3.1E-2)†	9.52E-1(1.3E-1)†	2.19E-1(1.1E-2)†
	10	**6.16E-2**(1.1E-3)	2.33E-1(7.2E-3)†	1.12E+0(2.4E-3)†	2.31E-1(6.1E-3)†	1.47E-1(6.9E-3)†	1.12E+0(5.5E-3)†	9.74E-1(1.0E-2)†	4.65E-1(7.0E-2)†	1.09E+0(9.3E-2)†	2.44E-1(6.1E-3)†

TABLE 6. Summary of the Wilcoxon rank-sum test results for the selected algorithms with respect to the mean IGD$^+$ value for MaF1 to MaF8.

	SMPSO	dMOPSO	MOPSOhv	MaPSO	MOEA/D	DBEA	NSGAIII	RVEA	ARMOEA
†	26	30	27	20	25	27	24	24	18
‡	6	2	5	12	6	5	6	8	11
=	0	0	0	0	1	0	2	0	3

†, ‡ and = denote the number of times the performance of the corresponding algorithm is significantly better, worse or has no significant difference with respect to that of the proposed algorithm, respectively.

has 3, 5, 8 and 10 objectives. Figure 18 shows that RVEA performs well in this problem.

MaF1 is a modified version of DTLZ1 which has an inverted Pareto optimal front which is linear. Regarding the mean IGD$^+$ value, our proposed algorithm performs better than SMPSO, dMOPSO, MOPSOhv, MOEA/D, DBEA and RVEA when the problem is scaled from 3 to 10 objectives, while MaPSO outperforms our proposed algorithm. Regarding NSGA-III, its performance is worse than that of our proposed algorithm when the problem has 3 and 5 objectives. For ARMOEA, it outperforms our proposed algorithm when the problem has 3 and 10 objectives.

MaF2 is a modified version of DTLZ2 which has a concave Pareto optimal front. Regarding the mean IGD$^+$ value, MaPSO is the winner in this test problem as it obtains the smallest IGD$^+$ values when the problem is scaled from 3 to 10 objectives. Our proposed algorithm outperforms SMPSO and MOPSOhv for MaF2-3 but it is outperformed when the number of objectives is 5, 8 and 10. Although dMOPSO, MOEA/D, DBEA, NSGA-III and RVEA outperform the proposed algorithm for MaF2-3, the performance of the proposed algorithm improves when the problem has 10 objectives. For ARMOEA, it outperforms the proposed algorithm when the problem has 3, 5 and 8 objectives, while the two algorithms have a similar performance when the problem has 10 objectives.

MaF3 is multimodal. It is a modified version of DTLZ3 but with a convex Pareto optimal front. Regarding the mean IGD$^+$ value, MOPSO-HGLSS performs better than SMPSO, dMOPSO, MOPSOhv, DBEA and NSGA-III when it was scaled from 3 to 10 objectives. Although the performance of MOPSO-HGLSS is worse than that of MaPSO for MaF3-3, it outperforms them when the problem has 5, 8 and 10 objectives. Furthermore, it should be noticed that MOEA/D, RVEA and ARMOEA perform better for MaF3-10.

MaF4 is multimodal. It is a modified of DTLZ3. Its Pareto optimal front is inverted and badly scaled. Regarding the mean IGD$^+$ value, our proposed algorithm outperforms the others when the problem has 3, 5, 8 and 10 objectives.

MaF5 is modified from DTLZ4. Its Pareto optimal front is convex and badly scaled. Similar to MaF4, our proposed algorithm outperforms the others when the problem has 3, 5, 8 and 10 objectives in terms of mean IGD$^+$ value.

MaF6 has a degenerate Pareto optimal front. Regarding the mean IGD^+ value, our proposed algorithm outperforms dMOPSO, while NSGA-III, RVEA and ARMOEA outperform our proposed algorithm. Although SMPSO, MOPSOhv, MaPSO, MOEA/D and DBEA outperform our proposed algorithm for MaF6-3 and MaF6-5, our proposed algorithm outperforms them when the problem has 10 objectives.

MaF7 has a disconnected Pareto optimal front. Regarding the mean IGD^+ value, our proposed algorithm outperforms the others when the problem has 3, 5, 8 and 10 objectives.

The Pareto optimal region of MaF8 in decision space is a 2D manifold, which allows a direct observation of the search behavior of a multi-objective optimization algorithm. Regarding the mean IGD^+ value, our proposed algorithm outperforms the others when the problem has 3, 5, 8 and 10 objectives.

Table 4 summarizes the results of the comparisons performed for DTLZ1, DTLZ2, and WFG1 to WFG9 in terms of IGD^+. With respect to IGD^+, MOPSO-HGLSS obtained better results in 275 out of 396 performance comparisons. Table 6 summarizes the results of the comparisons performed for MaF1 to MaF8 in terms of IGD^+. With respect to IGD^+, MOPSO-HGLSS obtained better results in 221 out of 288 performance comparisons. From these results, we conclude that MOPSO-HGLSS performed better that the other nine algorithms in terms of convergence and diversity because the two mechanisms included in HGLSS could properly balance the convergence and diversity of the MOPSO, which are two basic and very important issues in MOPs. In addition, MOPSO-HGLSS showed a promising performance as the number of objectives of the problems increased.

V. CONCLUSION AND FUTURE WORK

Most multi-objective particle swarm optimizers encounter difficulties when solving problems with more than three objectives. The reason is that the diluted selection pressure caused by the single global leader selection strategy of MOPSOs as the number of objectives increases has a negative effect on convergence and diversity, which are the two main goals for generating a proper set of solutions. Based on this observation, we proposed here a new algorithm called multi-objective particle swarm optimizer with hybrid global leader selection strategy (MOPSO-HGLSS). HGLSS has two global leader selection mechanisms: one is called Euclidean Distance Strategy (EDS) and the other one is called Space Expanding Strategy (SES). These two mechanisms aims to enhance the convergence and the diversity of the MOPSO, respectively. Performance investigation is conducted to facilitate the use of HGLSS which aims at balancing the trade-off between convergence and diversity of the MOPSO during the search. In addition, four MOPSOs (SMPSO, dMOPSO, MOPSOhv and MaPSO) and five popular MOEAs called MOEA/D, NSGA-III, DBEA, RVEA and ARMOEA were used to assess the performance of our proposed approach in terms of IGD^+. Supported by statistical tests, our performance investigation shows that HGLSS can properly balance the trade-off between convergence and diversity of our MOPSO. Thus, MOPSO-HGLSS outperforms the other algorithms in 19 MOPs (with 3, 5, 8 and 10 objectives for each selected problem) and has a promising performance in many-objective optimization problems.

Our future work will focus on investigating the performance of MOPSO-HGLSS in real-world many-objective problems. We are also interested in studying the impact of both mutation operators and mutation rates on MOPSO-HGLSS in greater depth.

REFERENCES

[1] H. Li, Y.-S. Ong, M. Gong, and Z. Wang, "Evolutionary multitasking sparse reconstruction: Framework and case study," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 733–747, Oct. 2019.

[2] M.-F. Leung and J. Wang, "Minimax and biobjective portfolio selection based on collaborative neurodynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 1, 2020, doi: 10.1109/TNNLS.2019.2957105.

[3] W.-N. Chen, D.-Z. Tan, Q. Yang, T. Gu, and J. Zhang, "Ant colony optimization for the control of pollutant spreading on social networks," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4053–4065, Sep. 2020.

[4] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Sep. 2002.

[5] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.

[6] A. Menchaca-Méndez, E. Montero, L. M. Antonio, S. Zapotecas-Martínez, C. A. C. Coello, and M.-C. Riff, "A co-evolutionary scheme for multi-objective evolutionary algorithms based on ϵ-dominance," *IEEE Access*, vol. 7, pp. 18267–18283, 2019.

[7] M.-F. Leung and S.-C. Ng, "A hybrid algorithm based on MOEA/D and local search for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[8] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 2419–2426.

[9] O. Schutze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 444–455, Aug. 2011.

[10] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.

[11] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[12] J. Shi, Q. Zhang, and J. Sun, "PPLS/D: Parallel Pareto local search based on decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1060–1071, Mar. 2020.

[13] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, Dec. 2011.

[14] H. Ishibuchi, Y. Hitotsuyanagi, H. Ohyanagi, and Y. Nojima, "Effects of the existence of highly correlated objectives on the behavior of MOEA/D," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2011, pp. 166–181.

[15] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[16] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

[17] A. Menchaca-Mendez, E. Montero, and S. Zapotecas-Martinez, "An improved S-Metric selection evolutionary multi-objective algorithm with adaptive resource allocation," *IEEE Access*, vol. 6, pp. 63382–63401, 2018.

[18] K. Bringmann, "Bringing order to special cases of klee's measure problem," in *Proc. Int. Symp. Math. Found. Comput. Sci.* Berlin, Germany: Springer, 2013, pp. 207–218.

[19] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.

[20] Q. Yang, W.-N. Chen, T. Gu, H. Zhang, J. D. Deng, Y. Li, and J. Zhang, "Segment-based predominant learning swarm optimizer for large-scale optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2896–2910, Sep. 2017.

[21] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[22] F. Li, J. Liu, S. Tan, and X. Yu, "R2-MOPSO: A multi-objective particle swarm optimizer based on R2-indicator and decomposition," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 3148–3155.

[23] B. Cao, J. Zhao, Z. Lv, X. Liu, S. Yang, X. Kang, and K. Kang, "Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization," *IEEE Access*, vol. 5, pp. 8214–8221, 2017.

[24] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. C. Coello, F. Luna, and E. Alba, "Multi-objective particle swarm optimizers: An experimental comparison," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2009, pp. 495–509.

[25] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based Metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Milti-Criteria Decision-Making*, Mar. 2009, pp. 66–73.

[26] A. Britto and A. Pozo, "Using reference points to update the archive of MOPSO algorithms in many-objective optimization," *Neurocomputing*, vol. 127, pp. 78–87, Mar. 2014.

[27] U. K. Wickramasinghe and X. Li, "Using a distance metric to guide PSO algorithms for many-objective optimization," in *Proc. 11th Annu. Conf. Genetic Evol. Comput. GECCO*, 2009, pp. 667–674.

[28] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Cham, Switzerland: Springer, 2015, pp. 110–125.

[29] S. Zapotecas Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proc. 13th Annu. Conf. Genetic Evol. Comput. GECCO*, 2011, pp. 69–76.

[30] I. C. Garcia, C. A. C. Coello, and A. Arias-Montano, "MOPSOhv: A new hypervolume-based multi-objective particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 266–273.

[31] Y. Xiang, Y. Zhou, Z. Chen, and J. Zhang, "A many-objective particle swarm optimizer with leaders selected from historical solutions by using scalar projections," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2209–2222, May 2020.

[32] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[33] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using Reference-Point-Based nondominated sorting approach—Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[34] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[35] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[36] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, Aug. 2018.

[37] C. R. Raquel and P. C. Naval, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proc. Conf. Genetic Evol. Comput. GECCO*, 2005, pp. 257–264.

[38] W. Hu and G. G. Yen, "Density estimation for selecting leaders and mantaining archive in MOPSO," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 181–188.

[39] M.-F. Leung, S.-C. Ng, C.-C. Cheung, and A. K. Lui, "A new strategy for finding good local guides in MOPSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1990–1997.

[40] J. Liu, F. Li, X. Kong, and P. Huang, "Handling many-objective optimisation problems with r2 indicator and decomposition-based particle swarm optimiser," *Int. J. Syst. Sci.*, vol. 50, no. 2, pp. 320–336, Jan. 2019.

[41] J. Branke and S. Mostaghim, "About selecting the personal best in multi-objective particle swarm optimization," in *Parallel Problem Solving From Nature-PPSN IX*. Berlin, Germany: Springer, 2006, pp. 523–532.

[42] N. Padhye, J. Branke, and S. Mostaghim, "Empirical comparison of MOPSO methods–guide selection and diversity preservation," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 2516–2523.

[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[44] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evol. Comput. CEC*, Oct. 2002, pp. 1051–1056.

[45] N. Al Moubayed, A. Petrovski, and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, Mar. 2014.

[46] C.-N. Lin, C.-L. Huo, S.-Y. Lin, Y.-H. Yu, and T.-Y. Sun, "Taguchi-based disturbance with tournament selection to improve on MOPSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1805–1810.

[47] M.-F. Leung, S.-C. Ng, C.-C. Cheung, and A. K. Lui, "A new algorithm based on PSO for multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 3156–3162.

[48] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Proc. Genetic Evol. Comput. Conf.* Berlin, Germany: Springer, 2003, pp. 37–48.

[49] H. Hajipour, H. B. Khormuji, and H. Rostami, "ODMA: A novel swarm-evolutionary Metaheuristic optimizer inspired by open source development model and communities," *Soft Comput.*, vol. 20, no. 2, pp. 727–747, Feb. 2016.

[50] L. Tang and X. Wang, "A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 20–45, Feb. 2013.

[51] K. Deb, *Optimization using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.

[52] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[53] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Proc. 3rd Mex. Int. Conf. Artif. Intell. (MICAI)* in Lecture Notes in Artificial Intelligence, vol. 2972, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, Eds. Berlin, Germany: Springer, Apr. 2004, pp. 688–697.

[54] M.-F. Leung and J. Wang, "A collaborative neurodynamic approach to multiobjective optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5738–5748, Nov. 2018.

[55] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

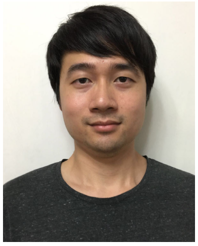
[56] H. Ishibuchi, H. Masuda, and Y. Nojima, "A study on performance evaluation ability of a modified inverted generational distance indicator," in *Proc. Genetic Evol. Comput. Conf. GECCO*, 2015, pp. 695–702.

[57] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005, pp. 105–145.

[58] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[59] G. W. Corder, *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Hoboken, NJ, USA: Wiley, 2009.

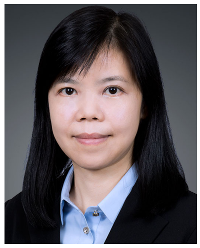
[60] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, Mar. 2017.



MAN-FAI LEUNG (Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2019. He is currently with the School of Science and Technology, The Open University of Hong Kong. His current research interests include portfolio optimization, multiobjective optimization, and computational intelligence.

CARLOS ARTEMIO COELLO COELLO (Fellow, IEEE) received the Ph.D. degree in computer science from Tulane University, USA, in 1996. Since 2010, he has been a Full Professor with the Department of Computer Science, CINVESTAV-IPN, Mexico City, Mexico. He specializes on the design of metaheuristics for solving nonlinear multi-objective problems. He currently has over 500 publications which, according to Google Scholar, report over 52 400 citations (with an H-index of 91). He received several awards, including the National Research Award in 2007 from the Mexican Academy of Science (in the area of exact sciences), the 2012 National Medal of Science in Physics, Mathematics, and Natural Sciences from Mexico's presidency (this is the most important award that a scientist can receive in Mexico), the prestigious 2013 IEEE Kiyo Tomiyasu Award, for pioneering contributions to single- and multiobjective optimization techniques using bioinspired metaheuristics, and the 2016 The World Academy of Sciences (TWAS) Award in Engineering Sciences. He is also an Associate Editor of several journals, including the two most prestigious in his area: IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and *Evolutionary Computation*.

CHI-CHUNG CHEUNG (Senior Member, IEEE) was born in Hong Kong. He received the B.Sc. and M.Phil. degrees from the City University of Hong Kong, Hong Kong, in 1991 and 1994, respectively, and the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 1998. He is currently with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include localization and machine learning.



SIN-CHUN NG (Senior Member, IEEE) received the B.Sc. degree (Hons.) in information technology and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, in 1990 and 2000, respectively. She is currently a Professor with the School of Science and Technology, The Open University of Hong Kong. Her research interests include evolutionary computation, neural networks, multimedia technology, and educational technology.

ANDREW KWOK-FAI LUI (Member, IEEE) received the B.Sc. degree in computer science and physics from The University of Sydney, Australia, in 1992, and the Ph.D. degree from the Research School of Information and System Engineering, The Australian National University, Canberra, ACT, Australia, in 1998. He is currently a Professor with the Department of Technology, The Open University of Hong Kong. His current research interests include computational intelligence, multi-objective optimization, traffic modeling, and computer science education.