Complex & Intelligent Systems (2020) 6:263–274
https://doi.org/10.1007/s40747-020-00134-7

ORIGINAL ARTICLE

A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems

Shufen Qin[1] · Chaoli Sun[1] · Guochen Zhang[1] · Xiaojuan He[2] · Ying Tan[1]

Received: 14 January 2020 / Accepted: 16 February 2020 / Published online: 13 March 2020

Abstract
Many evolutionary algorithms have been proposed for multi-/many-objective optimization problems; however, the tradeoff of convergence and diversity is still the challenge for optimization algorithms. In this paper, we propose a modified particle swarm optimization based on decomposition framework with different ideal points on each reference vector, called MPSO/DD, for many-objective optimization problems. In the MPSO/DD algorithm, the decomposition strategy is used to ensure the diversity of the population, and the ideal point on each reference vector can draw the population converge faster to the optimal front. The position of each individual will be updated by learning the demonstrators in its neighborhood that have less distance to the ideal point along the reference vector. Eight state-of-the-art evolutionary multi-/many-objective optimization algorithms are adopted to compare the performance with MPSO/DD for solving many-objective optimization problems. The experimental results on seven DTLZ test problems with 3, 5, 8, 10, 15 and 20 objectives, respectively, show the efficiency of our proposed method on solving problems with high-dimensional objective space.

Keywords Many-objective optimization · Decomposition · Different ideal points

Introduction

Multi-objective optimization problems (MOP) are widely involved in the real-world applications, for example, industrial scheduling [21], software engineering [19], and control system design [10]. The mathematical models of the multi-objective optimization problems are given as follows:

$$\begin{aligned} &\min \ \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ &s.t. \ \ \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in \Re^D$ is a solution in the D-dimensional decision space, $F : \Re^D \rightarrow \Re^m$ consists m objective functions $f_i(\mathbf{x}), i = 1, 2, \ldots, m$, and $\Re^m$ denotes the m-dimensional objective space. In general, due to the conflicting nature of the objectives, no solution can be the optimum of all objective functions simultaneously, instead, a set of trade-off solutions, called Pareto optimal solutions or non-dominant solution set [7], will be found for the optimization problem. The set of all Pareto-optimal solutions is called the Pareto set (PS) and its mapping to the objective space is the Pareto front (PF).

Different evolutionary multi-objective optimization (EMO) methods have been proposed for solving multi-objective optimization problems [1,2,9,31]. Especially, in recent years, optimization methods for problems with more than three objectives, which are called many-objective optimization problems (MaOPs), have been obtained more and more attentions because the performances of canonical algorithms for multi-objective problems will be degraded much quickly with the number of objective increases. Generally,

✉ Chaoli Sun
chaoli.sun.cn@gmail.com

Shufen Qin
qin980519@sina.com

Guochen Zhang
imzgc@hotmail.com

Xiaojuan He
hxj701104@163.com

Ying Tan
tanying2015@163.com

1 Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

2 College of Applied Science, Taiyuan University of Science and Technology, Taiyuan 030024, China

the approaches proposed for solving MaOPs can be roughly classified into three categories.

The first category is multi-/many-objective optimization algorithms based on dominance relationship. The most representative one for multi-objective problems is NSGA-II [9], which was proposed by Deb in 2002. However, the performance of NSGA-II will be deteriorated when the number of objective increases because of the loss the selection pressure. Therefore, scholars have focused on finding more and more efficient strategies on dominance-based evolutionary algorithms for solving many objective optimization problems, such as ε-dominance [12], θ-dominance [30], and fuzzy-Pareto-dominance [23]. Yang et al. [27] proposed a grid-based many-objective evolutionary algorithm (GrEA), in which grid domination and grid difference were used to improve the selection pressure. Zhang et al. [32] proposed a many-objective evolutionary algorithm based on knee point (KnEA), in which the distance between hyperplane and a knee point was used to select better non-dominant solutions, which greatly improves the selection pressure.

The second category is multi-/many-objective optimization algorithms based on decomposition strategy, which can further be divided into two types, one is that the multi-/many-objective optimization problems are transformed to a set of single-objective optimization problems [14,15,25,29,31], and the other is that the complex multi-objective algorithms are transformed to a set of simple multi-objective optimization problems [8,18]. In [26], Xiang et al. proposed a vector angle-based many-objective evolutionary algorithm (VaEA) which uses maximum-vector-angle-first principle and worse-elimination principle to maintain the diversity and convergence of the population. Cheng et al. [4] proposed a many-objective evolutionary algorithm guided by a set of reference vectors (RVEA) and Jiang et al. [11] proposed a many-objective evolutionary algorithm based on reference direction (SPEA/R).

The indicator-based evolutionary algorithms fall into the third category, in which the performance indicator is used instead of fitness to select individuals. Zitzler and kunzli proposed the indicator-based evolutionary algorithm (IBEA) [33], in which a binary performance measure was proposed in the selection process. Bader et al. [1] proposed a hypervolume estimation algorithm, called HypE, for many-objective optimization, in which the Monte Carlo simulation was utilized to approximate the exact hypervolume values. Tian et al. [22] proposed an indicator-based multi-objective evolutionary algorithm with reference point adaptation (AR-MOEA), in which the algorithm adjusted the position of reference points based on the contribution of the indicator to improve the performance of irregular Pareto frontier problems.

In recent years, there are also other algorithms that combine the above three strategies. Li et al. [13] proposed a MOEA/DD algorithm, in which both decomposition and dominance strategies were utilized. Based on the performance indicator and domination relationship, Wang et al. [24] proposed a Two-arch2 algorithm. Deb and Jain [8] extended the well-known NSGA-II and proposed the NSGA-III algorithm to deal with many-objective optimization problem, in which a set of reference points were utilized to maintain the diversity of the population during the search with non-dominated sorting mechanism.

Literature reviews show that there are a small number of algorithms with the PSO framework proposed for solving many-objective optimization problems. The reason, we analyze, is because of the quick convergence of the PSO algorithm, which may not be able to provide a good diversity for finding the optimal Pareto front. In this paper, a modified particle swarm optimization with decomposition strategy and different ideal points, called MPSO/DD, will be proposed, in which the decomposition strategy is adopted to ensure the uniformity of the final outputs, and multiple ideal points are utilized to drive the population to quickly convergence to the optimal front. The learning strategy proposed by Cheng and Jin [3] is adopted to update the position of each individual, in which the demonstrators are those with less distance to the ideal point along the reference vector.

The paper is organized as follows: Section 2 describes our proposed method in detail. Experimental results are given in Section 3 with some discussions. Finally, Section 4 gives the conclusions and talks about some work we can do in the future.

The proposed MPSO/DD

Overall framework

Algorithm 1 gives the pseudocode of our proposed MPSO/DD algorithm. A series of reference vector $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{i,m}), i = 1, 2, \ldots, N$ will be generated in the objective space at first. Then a population, each individual in which has its own position $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), i = 1, 2, \ldots, N$ and velocity $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD}), i = 1, 2, \ldots, N$, will be generated in the upper and lower bounds, and evaluated using the objective functions. All non-dominated solutions in the population will be saved to the archive Arc. If the stopping criteria is not met, the following steps will be repeated. Determine the ideal point for each reference vector using all non-dominated solutions. Sort the Tchebycheff values of an individual on all reference vectors in an ascending order, find the first reference vector after the sorting which has not been associated with any individual, and assign this reference vector with the current individual. Therefore, each individual will be associated with one and only one reference vector. After that, neighbors of each individual will be used to update the position of this individual, and correspondingly a new offspring population will be generated. Next, a new parent population will be selected from the parent and offspring populations according to the environmental selection strategy proposed in [20]. Finally, all non-dominated solutions stored in the external archive will be updated using the current population obtained by environmental selection and output when the terminal condition is satisfied, which can be seen in Step 13 of Algorithm 1.

In the following, we will give a detailed description on main parts of Algorithm 1:

Algorithm 1 MPSO/DD(P,N,T,t_{max})

```
Require:
    $P$: population;
    $N$: the size of population;
    $T$: the number of individuals in the neighborhood;
    $t_{max}$: the maximum number of iteration;
Ensure:
    Archive $Arc$;
1: /* Initialization*/ ;
2: Set $Arc = \phi$;
3: Initialize the reference vector set $\{\lambda_i = (\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,m}), i = 1, 2, \ldots, N\}$;
4: Initialize the position and velocity of each individual in the population $P$, and calculate the objective values for each individual in $P$;
5: Save all non-dominated solutions in $P$ to the archive $Arc$;
6: /* Main loop*/;
7: while $t \le t_{\max}$ do
8:     Update the ideal point for each reference vector; (Refer to Algorithm 2)
9:     Associate the reference vector to an individual;
10:    Find neighbor individuals to each individual;
11:    Generate a new offspring population; (Refer to Algorithm 3)
12:    Environmental selection; (Refer to Algorithm 4)
13:    Update the archive $Arc$; (Refer to Algorithm 5)
14:    $t = t + 1$;
15: end while
```

Ideal point generation

Different from decomposition-based methods proposed previously where only one ideal point is used in the whole evolution, in our method, each reference vector has its own ideal point, which was determined by the objective values of individuals in the non-dominated archive, to speed up the convergence along the reference vector. Figure 1 gives a simple example to show our strategy to generate the ideal point for each reference vector. In Fig. 1, given an arbitrary reference vector λ_i, the circles in red represent the non-dominated individuals in Arc, and the circle in yellow is the ideal point which has the minimum distance among five non-dominated individuals along the reference vector λ_i to the origin. Equation (2) gives the method to calculate the distance of each individual in Arc along the reference vector to the origin.

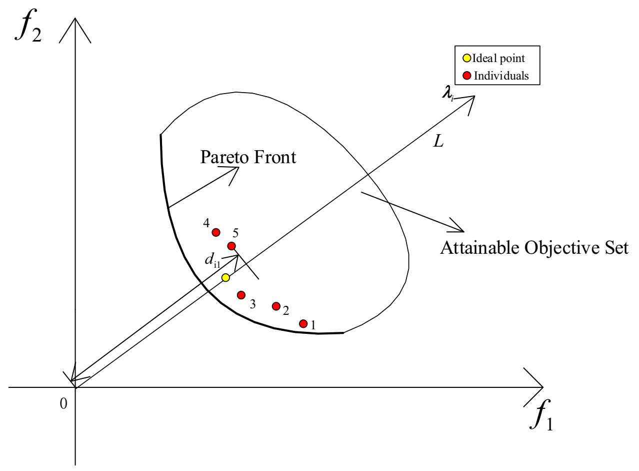

Fig. 1 An example to show the ideal point setting

$$d_{ij} = \frac{||\mathbf{F}_{norm}(\mathbf{x}_j)^T \lambda_i||}{||\lambda_i||}, \ j = 1, 2, \ldots, K, \tag{2}$$

where

$$F_{norm,k}(\mathbf{x}_j) = \frac{f_k(\mathbf{x}_j) - f_{\min,k}}{f_{\max,k} - f_{\min,k}}. \tag{3}$$

In Eqs. (2) and (3), $\mathbf{F}_{norm} = (\mathbf{F}_{norm,1}, \mathbf{F}_{norm,2}, \ldots, \mathbf{F}_{norm,m})$ is the objective vector after normalization, λ_i refers to the current reference vector. $f_{\max,k}$ and $f_{\min,k}$ are the maximum and minimum objective values on kth objective in the non-dominated solution set Arc, respectively. K is the size of the current non-dominated archive Arc.

Algorithm 2 gives the pseudocode of the determination of the ideal points. In Algorithm 2, $|Arc|$ and $|\lambda|$ represent the number of non-dominated solutions in the archive Arc and the number of reference vectors, respectively. The distance between the point, projected by a non-dominated solution in the archive Arc on the reference vector λ_i, and the origin will be calculated. The point with minimal distance to the origin along the reference vector will be the ideal point of this reference vector.

The offspring generation

In the original social learning particle swarm optimization proposed by Cheng and Jin [3], the velocity and position of each individual are updated as follows:

$$v_{ij}(t+1) = r_1 v_{ij}(t) + r_2(x_{wj}(t) - x_{ij}(t)) + r_3(\bar{x}_j(t) - x_{ij}(t)) \tag{4}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \tag{5}$$

where v_{ij} and x_{ij} are the jth velocity and position of individual i, respectively. x_{wj} is the jth position of individual w

Algorithm 2 The determination of ideal point for each reference vector

```
Require:
    $Arc$: a set of non-dominated solutions;
    $\lambda$: a set of the reference vectors;
    $\mathbf{f}_{norm}$: the normalized objective values;
Ensure:
    {IP}: an ideal point set;
1: for $i = 1 : |\lambda|$ do
2:     for $j = 1 : |Arc|$ do
3:         $d_{ij} = \frac{\|\mathbf{f}_{norm}(\mathbf{x}_j)^T \lambda_i\|}{\|\lambda_i\|}$;
4:     end for
5:     $mind_i = \min\{d_{ij}, j = 1, 2, \ldots, K\}$ ;
6:     $\mathbf{IP_i} = mind_i \lambda_i$;
7: end for
```

whose fitness is better than individual i. $\bar{x}_j$ is the mean position of the current population on jth dimension. r_1, r_2 and r_3 are random numbers generated uniformly between 0 and 1. The original social learning particle swarm optimization was proposed for single-objective problems and has been shown a good performance to find better optimal solutions especially on large-scale optimization because of its good diversity. However, as we know, in the multi-/many-objective optimization, normally the individuals do not dominate each other and it is difficult to tell which individual is better than another based on the objective values, especially when the number of objectives increase. Therefore, in our method, for an individual i, we first calculate the distances between individuals in the neighborhood and the ideal point of the reference vector individual i associated with (shown in Eq. (6)). In Eq. (6), $\mathbf{F}(\mathbf{x}_j)$, $j = 1, 2, \ldots, |NI|$ the objective vector of individual j in the neighborhood of individual i, $|NI|$ is the number of neighbors of individual $\mathbf{x}_j$, $\mathbf{IP}_i$ is the ideal point of reference vector λ_i, and $d1_{i,j}$ is the distance between individual j and the origin along the reference vector λ_i. All distances will be sorted in a descent order, and correspondingly, individual i can learn from those neighbors who has better convergence to the Pareto front. Both Eqs. (7) and (8) are used for updating the velocity of an individual with probabilities to prevent the population from falling into local optima. As we know, the convergence speed of the social learning particle swarm optimization algorithm is limited because of its good diversity; therefore, the coefficient proposed in [5], i.e., 0.729, is utilized in Eq. (7) to speed up the convergence. In Eq. (7), x_{wj} represents the jth dimension of individual w whose distance to the original along with the current reference vector is better than individual i. r_1 and r_2 are random number generated uniformly between 0 and 1. Equation (8) is used to randomly initialize the velocity so as to jump out of the local optimal position.

$$d1_{i,j} = ||\mathbf{F}(\mathbf{x}_j) - \mathbf{IP}_i|| \tag{6}$$

$$v_{ij} = 0.729 * (r_1 * v_{ij} + r_2 * (x_{wj} - x_{ij})) \tag{7}$$

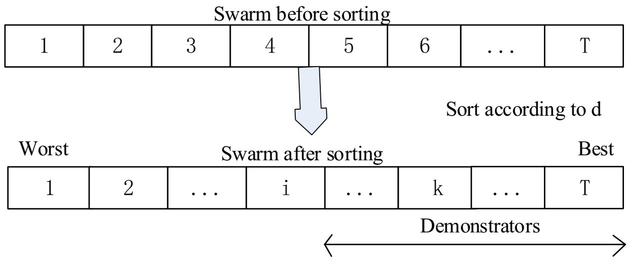

Fig. 2 Demonstration selection

$$v_{ij} = r_1 * (v_{j,\max} - v_{j,\min}) + v_{j,\min}. \tag{8}$$

Algorithm 3 gives the pseudocode of the generation of an offspring. For each reference vector, the distance to the ideal point of its neighbor individuals will first be calculated, and sorted in a descending way. Figure 2 gives a simple example to show how to select the demonstrator according to the distance to the ideal point. The best position on the right hand is the individual that has the minimal distance to the ideal point along the reference vector. A threshold, 0.99, is given empirically in line 7 of Algorithm 3, for determining which equation is to be used for velocity updating. To see the efficiency of parameter settings, we conducted three cases of empirical experiments on DTLZ3 with different number of objectives:

Case1: Without the coefficient 0.729 in Eq. (7).

Case2: Only Eq. (7) is utilized in the proposed method.

Case3: The threshold using Eq. (8) is set to a half of 0.99, i.e. 0.495.

Table 1 gives the results of three cases as well as our proposed setting. From Table 1, we can see that the results obtained in Case2 and Case3 are all worse than those obtained by MPSO/DD, which shows that 0.99 is best to be the threshold to select equation for velocity updating. Compared to Case1, we can see that our proposed MPSO/DD obtained better or competitive results on DTLZ3 problem with more than 10 objectives, which shows that the coefficient 0.729 is significant for the convergence of the algorithm to optimize the problems with high-dimensional objectives.

The environmental selection

After the objective evaluation of each offspring, both parent and offspring individuals will be combined together and calculated the CAD proposed in [20] on each reference vector, where

$$CAD(i, j) = \frac{\cos\theta_{i,j}}{d1_{i,j}}. \tag{9}$$

In Eq. (9), $\cos\theta_{i,j}$ represents cosine of the angle between the ith reference vector and the jth individual in the combination population of parent and offspring. The larger the

Table 1 The statistical results (mean and standard deviation) of the IGD values obtained by three cases and MPSO/DD on DTLZ3

M	Case1	Case2	Case3	MPSO/DD
3	5.9744e+00 (3.23e+00)≈	8.9132e+01 (4.15e+01)+	1.6338e+02 (8.93e+00)+	**4.8332e+00 (1.49e+00)**
5	4.1979e+00 (2.70e+00)≈	9.1218e+01 (2.79e+01)+	1.6889e+02 (9.47e+00)+	**3.6665e+00 (1.45e+00)**
8	**2.6646e+00 (1.67e+00)**–	8.5275e+01 (3.54e+01)+	1.7035e+02 (1.52e+01)+	3.2763e+00 (1.24e+00)
10	4.2037e+00 (6.08e+00)≈	7.8929e+01 (3.30e+01)+	1.6695e+02 (1.57e+01)+	**3.2172e+00 (1.91e+00)**
15	7.8629e+00 (2.33e+01)≈	8.9359e+01 (2.47e+01)+	1.6888e+02 (1.23e+01)+	**3.0568e+00 (1.16e+00)**
20	4.2476e+01 (4.99e+01)+	9.0674e+01 (4.03e+01)+	1.7174e+02 (1.66e+01)+	**3.7659e+00 (2.33e+00)**

The best results are highlighted

Algorithm 3 The generation of an offspring population

```
Require:
    $P$: population;
    $T$: the number of individuals in the neighborhood;
1: /* Position updating of each individual*/ ;
2: for $i = 1 : N$ do
3:     Calculate the distance of each individual $j$ in the neighborhood of individual $i$ to the ideal point along the reference vector using Eq. (6);
4:     Sort the distances in an ascending order, and keep the individuals after sorting in $\mathbf{X}_i$;
5:     for $j = 1 : D$ do
6:         /* Velocity updating*/;
7:         if $rand() < 0.99$ then
8:             On each dimension, randomly select individual from $\mathbf{X}_i$ that has shorter distance to the origin than that of individual $i$.
9:             Updating velocity using Eq. (7);
10:        else
11:            Updating velocity using Eq. (8);
12:        end if
13:        /*Position updating*/ ;
14:        Updating position of each individual using Eq. (5);
15:    end for
16: end for
```

Algorithm 4 Environment selection

```
Require:
    $P(t)$ : the parent population; $Q(t)$ : the offspring population;
    {IP} : the ideal point set;
    $\lambda$: the reference vector set;
Ensure:
    $P(t+1)$ : a new parent population;
1: for $i = 1 : |\lambda|$ do
2:     for $j = 1 : 2 * |P(t)|$ do
3:         Calculate $CAD$ value using Eq. (9);
4:     end for
5:     Associate the individual with maximum $CAD(i, j), j = 1, 2, \ldots, 2 * |P(t)|$ to the reference vector $i$;
6: end for
7: $P(t+1)$ = all individuals assigned to reference vector set ;
```

$\cos\theta$ is (the smaller the angle is), the closer the individual and reference vector are, i.e. the even distribution of the individuals in the objective space. $d1_{i,j}$ is calculated using Eq. (6). It can be seen obviously that the larger the CAD value is, the better between the balance on diversity and convergence of individuals.

Algorithm 4 gives the pseudocode of the environment selection. Each individual in the parent and offspring population will be calculated the CAD value related to each reference vector, and the individual with maximum CAD value to each reference vector will be kept to the next generation.

The archive updating

All non-dominated solutions will be saved in the archive Arc. When new population is generated and evaluated on objective functions, they will be used to update the archive Arc. Algorithm 5 gives the pseudocode of the archive updating. In Algorithm 5, $Arc(t-1)$ represents the archive at the $t-1$th generation and P is the offspring population. Note that the size of the archive is fixed to the size of population N.

Algorithm 5 The archive updating

```
Require:
    $Arc(t-1)$;
    $P$: the offspring population;
Ensure:
    $Arc(t)$;
1: $Arc(t) = \phi$;
2: $A = Arc(t-1) \cup P$;
3: if $|A| > |P|$ then
4:     for $i = 1 : |P|$ do
5:         for $j = 1 : |A|$ do
6:             Calculate the $CAD(i, j)$ between individual $j$ and the reference vector $i$;
7:         end for;
8:         Sort the $CAD(i)$ in descending order, and the first individual will be saved in $Arc(t)$;
9:     end for;
10: else
11:     $Arc(t) = A$;
12: end if
```

In Algorithm 5, the offspring population P will first be combined together with the non-dominated individuals in the archive $Arc(t-1)$. If the size of $Arc(t)$ is larger than the size of population, we will keep the individual with the

Table 2 The parameter setting in the experiments

Problem	Number of objectives (M)	Number of variables (D)	Parameter (k)
DTLZ1	3, 5, 8, 10, 15, 20	$M-1+k$	$k=5$
DTLZ2	3, 5, 8, 10, 15, 20	$M-1+k$	$k=10$
DTLZ3	3, 5, 8, 10, 15, 20	$M-1+k$	$k=10$
DTLZ4	3, 5, 8, 10, 15, 20	$M-1+k$	$k=10$
DTLZ5	3, 5, 8, 10, 15, 20	$M-1+k$	$k=10$
DTLZ6	3, 5, 8, 10, 15, 20	$M-1+k$	$k=10$
DTLZ7	3, 5, 8, 10, 15, 20	$M-1+k$	$k=20$

Table 3 The number of reference vectors related to the number of objectives

Number of objectives (M)	The number of evaluations	Number of reference vector (λ)
3	30,000	153
5	30,000	210
8	30,000	156
10	30,000	220
15	30,000	120
20	30,000	210

maximum CAD value on the corresponding reference vector into $Arc(t)$. Otherwise, all individuals in $Arc(t-1)$ will be kept to $Arc(t)$.

Experimental results and discussion

Parameter setting

To verify the effectiveness of our proposed MPSO/DD algorithm on many-objective optimization problems, seven DTLZ test functions are selected and tested on 3, 5, 8, 10, 15 and 20 objectives, respectively. The obtained results are compared with those of NSGA-III, KnEA, RVEA, MOEA/DD, SPEAR, GrEA and BiGE [16] that are state-of-the-art algorithms for many-objective problems, and also compared with NMPSO [17], which is proposed for many-objective optimization based on PSO. The experimental results of these seven algorithms are run on PlatEMO proposed by Tian [28]. The parameters of MPSO/DD are given in Table 2. Also, the relationship between the number of objectives and the dimension of variables are given in Table 2.

Table 3 gives other parameters used in the experiments that related to different number of objectives, including the number of objective evaluations and correspondingly the size of reference vector set. The reference vectors are generated uniformly according to the strategy proposed in MOEA/D [31]. The number of population is consistent with the size of reference vectors. All other parameters needed to be set in our experiments are analyzed and given in Sect. 2.3. The parameters used in the comparison algorithms are set same as those used in the corresponding method.

Performance metrics

To compare the performance of different algorithms, the inverted generational distance (IGD) [6] is used as indicator to evaluate the performance of different algorithms. Suppose P^* is a set of points uniformly distributed on the optimal Pareto surface in the objective space and P is a set of non-dominated solutions, then the IGD value is defined as follows:

$$IGD(P, P^*) = \frac{\sum_{\mathbf{x}\in P^*} \min_{\mathbf{y}\in P} dist(\mathbf{x}, \mathbf{y})}{|P^*|}, \tag{10}$$

where $dist(\mathbf{x}, \mathbf{y})$ represents Euclidean distance between two positions $\mathbf{x}$ and $\mathbf{y}$. Therefore, the IGD is the average value of the minimum distance from each position in P^* to P, which is used to measure the convergence and diversity of the non-dominated solution set P that obtained. In our experiments, we selected 10,000 solutions from the real Pareto front P^* for comparison. The smaller the IGD value is, the better the P is.

Experimental result

Table 4 gives the statistical IGD results of the proposed MPSO/DD algorithm and the other seven algorithms on seven DTLZ test functions. The results of Wilcoxon rank sum test are also given: '+', '−' and '≈', respectively, indicate that the results of MPSO/DD algorithm are superior, inferior and similar to those of the comparative algorithm. The data in boldface in Table 4 represents the best results of all algorithms. All results are obtained on 20 independent running. From Table 4, we can clearly see that our proposed MPSO/DD method obtained better results on DTLZ 2, 3, 5 and DTLZ6 problems with high-dimensional objective space. Except the DTLZ2 with 20 objectives, MPSO/DD obtained better results on these problems with 10, 15, and

Table 4 The statistical results(mean and standard deviation) of the IGD values obtained by NSGA-III, KnEA, RVEA, MOEA/DD, SPEAR, GrEA, BiGE and MPSO/DD on DTLZ1 TO DTLZ7

M Problem	3	5	8	10	15	20
DTLZ1						
NSGA-III	3.5089e − 02 (8.50e − 02)−	2.4947e − 01 (4.12e − 01)−	4.1280e − 01 (3.77e − 01)−	9.8668e − 01 (9.58e − 01)+	9.4176e − 01 (1.47e + 00) ≈	5.8347e − 01 (6.43e − 01) ≈
KnEA	4.7610e − 02 (3.66e − 02)−	1.5089e − 01 (1.24e − 01)−	6.5931e + 00 (9.23e + 00)+	5.6285e + 00 (1.25e + 01)+	1.2229e + 01 (8.31e + 00)+	6.6802e + 00 (7.45e + 00)+
RVEA	8.4182e − 02 (2.46e − 01)−	1.6280e − 01 (2.87e − 01)−	2.0835e − 01 (3.84e − 01)−	1.7272e − 01 (1.10e − 01)−	1.9936e − 01 (6.97e − 02)−	2.1372e − 01 (1.46e − 01)−
MOEA/DD	6.9908e − 02 (2.40e − 01)−	**1.1731e-01 (2.82e-01)**−	**1.3848e-01 (1.72e-01)**−	**1.2950e-01 (1.01e-01)**−	**1.4302e-01 (4.79e-02)**−	**2.0566e-01 (3.93e-02)**−
SPEAR	6.3444e − 02 (1.72e − 01)−	4.6187e − 01 (1.27e + 00)+	1.0506e + 00 (1.55e + 00)+	3.9625e + 00 (3.63e + 00)+	3.3964e + 00 (6.40e + 00)+	6.1463e + 00 (6.95e + 00)+
GrEA	1.1682e − 01 (6.30e − 02) ≈	1.7914e − 01 (9.64e − 02)−	5.5160e − 01 (9.64e − 01)+	4.8043e + 00 (1.93e + 01) ≈	3.1042e + 01 (2.01e + 01)+	8.5951e + 00 (2.08e + 01)+
BiGE	**3.2131e-02 (1.65e-02)**−	1.5526e − 01 (1.18e − 01)−	8.7500e − 01 (5.33e − 01)+	2.7892e + 00 (2.55e + 00)+	2.9134e + 00 (4.17e + 00)+	2.8458e + 00 (2.08e + 00)+
MPSO/DD	2.3029e − 01 (1.91e − 01)	4.1405e − 01 (1.78e − 01)	4.5166e − 01 (1.29e − 01)	4.6162e − 01 (1.89e − 01)	6.1970e − 01 (2.87e − 01)	5.5316e − 01 (1.84e − 01)
DTLZ2						
NSGA-III	4.0978e − 02 (2.56e − 05)−	1.6597e − 01 (1.36e − 04)+	3.2515e − 01 (9.74e − 04)+	5.2042e − 01 (8.10e − 02)+	6.8347e − 01 (1.63e − 02)+	6.9578e − 01 (1.50e − 02)−
KnEA	5.1739e − 02 (1.75e − 03)+	1.7769e − 01 (1.26e − 03)+	3.8602e − 01 (7.17e − 03)+	4.5349e − 01 (5.39e − 03)+	6.2222e − 01 (3.21e − 03)+	6.1597e − 01 (2.22e − 03)−
RVEA	4.1421e − 02 (3.02e − 04)+	1.6542e − 01 (8.31e − 05)+	3.1847e − 01 (3.52e − 04)−	4.5827e − 01 (4.20e − 04)+	6.3037e − 01 (3.18e − 03)+	6.1953e − 01 (9.09e − 03)−
MOEA/DD	4.1086e − 02 (1.15e − 04)−	1.6534e − 01 (1.30e − 04)+	**3.1687e-01 (2.38e-04)**−	4.5965e − 01 (5.06e − 04)+	6.2469e − 01 (3.15e − 03)+	**6.1469e-01 (4.92e-03)**−
SPEAR	**4.1011e-02 (2.74e-05)**−	1.6620e − 01 (1.49e − 04)+	3.3393e − 01 (1.67e − 03)+	5.0019e − 01 (8.81e − 03)+	6.8838e − 01 (1.52e − 02)+	7.1837e − 01 (1.56e − 02)−
GrEA	5.6578e − 02 (6.35e − 04)+	1.7247e − 01 (7.74e − 04)+	3.5063e − 01 (2.20e − 03)+	4.2217e − 01 (1.05e − 03)+	6.2816e − 01 (2.93e − 02) ≈	6.1822e − 01 (1.20e − 03)−
BiGE	6.0816e − 02 (2.04e − 03)+	2.0661e − 01 (5.17e − 03)+	4.0479e − 01 (3.70e − 03)+	4.6530e − 01 (3.61e − 03)+	6.5473e − 01 (1.03e − 02)+	6.4504e − 01 (1.16e − 02)−
MPSO/DD	4.1172e − 02 (6.13e − 05)	**1.6113e-01 (3.21e-04)**	3.2029e − 01 (1.42e − 03)	**4.0232e-01 (1.83e-03)**	**6.0687e-01 (4.12e-02)**	7.8543e − 01 (4.84e − 02)
DTLZ3						
NSGA-III	1.5321e + 00 (6.82e − 01)−	5.5692e + 00 (2.84e + 00)+	1.1522e + 01 (5.17e + 00)+	2.8146e + 01 (1.93e + 01)+	3.0756e + 01 (2.71e + 01)+	2.0028e + 01 (2.59e + 01)+
KnEA	7.3467e − 01 (5.82e − 01)+	**1.6631e+00 (1.63e+00)**−	8.2251e + 01 (3.17e + 01)+	2.1250e + 02 (1.41e + 02)+	4.5056e + 02 (8.97e + 01)+	4.8183e + 02 (1.31e + 02)+
RVEA	3.1500e + 00 (1.57e + 00)+	5.4283e + 00 (1.24e + 00)+	4.7234e + 00 (1.62e + 00)+	7.3872e + 00 (2.61e + 00)+	4.2149e + 00 (3.06e + 00) ≈	5.9175e + 00 (9.89e − 01)+
MOEA/DD	2.2914e + 00 (1.29e + 00)+	6.3796e + 00 (2.13e + 00)+	3.6611e + 00 (1.34e + 00) ≈	1.1341e + 01 (6.00e + 00)+	4.3371e + 00 (1.61e + 00)+	5.4475e + 00 (1.66e + 00)+
SPEAR	2.3945e + 00 (7.75e − 01)+	1.0360e + 01 (3.50e + 00)+	7.2580e + 01 (2.32e + 01)+	1.1655e + 02 (3.31e + 01)+	1.8525e + 02 (3.37e + 01)+	1.9985e + 02 (3.81e + 01)+
GrEA	4.4394e − 01 (3.04e − 01)+	2.9414e + 00 (1.99e + 00) ≈	1.5937e + 01 (3.61e + 00)+	2.1999e + 01 (1.34e + 01)+	1.4799e + 02 (2.85e + 01)+	3.1145e + 02 (8.25e + 01)+
BiGE	1.4308e − 01 (2.14e − 01)+	1.8686e + 00 (8.38e − 01)−	5.5136e + 01 (1.35e + 01)+	7.4257e + 01 (1.43e + 01)+	4.6528e + 01 (1.51e + 01)+	5.3297e + 01 (1.39e + 01)+
MPSO/DD	**4.8332e+00 (1.49e+00)**	3.6665e + 00 (1.45e + 00)	**3.2763e+00 (1.24e+00)**	**3.2172e+00 (1.91e+00)**	**3.0568e+00 (1.16e+00)**	**3.7659e+00 (2.33e+00)**
DTLZ4						
NSGA-III	**4.0988e-02 (2.07e-05)**−	1.6644e − 01 (1.37e − 04)+	3.3713e − 01 (2.62e − 02)−	4.7374e − 01 (2.37e − 02)+	6.6855e − 01 (3.73e − 02)+	6.6036e − 01 (1.57e − 02)+
KnEA	5.3582e − 02 (1.15e − 03)+	1.8027e − 01 (3.48e − 03)+	3.7247e − 01 (4.26e − 03)+	4.6498e − 01 (2.76e − 03)+	6.2206e − 01 (2.44e − 03)−	6.3549e − 01 (7.30e − 03)+
RVEA	4.1728e − 02 (5.25e − 04)+	1.6558e − 01 (1.86e − 04)+	3.2380e − 01 (1.16e − 03)−	4.5819e − 01 (2.85e − 04)+	6.4281e − 01 (8.78e − 03)+	6.2379e − 01 (1.44e − 04) ≈
MOEA/DD	4.1153e − 02 (1.11e − 04)+	1.6551e − 01 (3.99e − 05)+	**3.1887e-01 (8.86e-04)**−	4.7041e − 01 (2.12e − 02)+	6.4093e − 01 (1.99e − 02)+	6.2322e − 01 (2.48e − 04) ≈
SPEAR	4.1116e − 02 (7.12e − 05)+	1.6693e − 01 (4.45e − 04)+	3.6159e − 01 (3.33e − 03)+	5.1076e − 01 (1.07e − 02)+	6.6929e − 01 (5.42e − 03)+	7.2435e − 01 (2.12e − 02)+
GrEA	5.9456e − 02 (2.40e − 03)+	1.7359e − 01 (2.19e − 03)+	3.5120e − 01 (1.14e − 03)−	**4.2480e-01 (7.05e-04)**−	**5.9052e-01 (1.02e-03)**−	6.5066e − 01 (1.18e − 02)+

Table 4 continued

M Problem	3	5	8	10	15	20
BiGE	5.9751e − 02 (2.04e − 03)+	2.0923e − 01 (3.65e − 03)+	4.0097e − 01 (4.75e − 03)+	6.2444e − 01 (4.00e − 02)+	6.4345e − 01 (5.72e − 03)+	8.0485e − 01 (5.35e − 03)+
MPSO/DD	4.1262e − 02 (6.58e − 05)	**1.6487e-01 (6.48e-04)**	3.5425e − 01 (5.54e − 03)	4.5638e − 01 (9.80e − 03)	6.2425e − 01 (3.95e − 03)	**6.2293e-01 (3.72e-03)**
DTLZ5						
NSGA-III	7.6515e − 03 (6.04e − 04)+	1.3321e − 01 (3.33e − 02)+	3.1124e − 01 (1.42e − 01)+	3.0480e − 01 (5.75e − 02)+	3.4934e − 01 (3.34e − 02)+	3.6017e − 01 (3.16e − 02)+
KnEA	**5.6052e-03 (7.40e-04)**+	1.8213e − 01 (2.71e − 02)+	2.5903e − 01 (8.17e − 02)+	3.6714e − 01 (5.79e − 02)+	5.9636e − 01 (1.59e − 01)+	6.1589e − 01 (2.00e − 01)+
RVEA	6.1438e − 02 (5.64e − 03)+	1.9670e − 01 (2.48e − 02)+	4.8030e − 01 (1.15e − 01)+	2.2531e − 01 (7.62e − 02)+	2.0077e − 01 (2.43e − 02)+	2.1355e − 01 (8.74e − 02)+
MOEA/DD	2.3495e − 02 (2.87e − 04)+	1.1221e − 01 (9.11e − 03)+	1.3710e − 01 (4.15e − 02) ≈	1.3518e − 01 (3.25e − 03) ≈	1.9245e − 01 (9.99e − 04)+	1.9331e − 01 (3.68e − 04) ≈
SPEAR	2.4208e − 02 (5.33e − 04)+	1.8850e − 01 (2.02e − 02)+	3.7886e − 01 (7.30e − 02)+	8.3605e − 01 (2.14e − 01)+	1.1021e + 00 (2.72e − 01)+	1.0483e + 00 (2.18e − 01)+
GrEA	1.9725e − 02 (8.39e − 04)+	1.6018e − 01 (4.80e − 02)+	2.5147e − 01 (6.53e − 02)+	3.0523e − 01 (8.17e − 02)+	5.3713e − 01 (6.34e − 02)+	8.1883e − 01 (8.21e − 02)+
BiGE	9.8871e − 03 (8.30e − 04)+	8.4528e − 02 (7.57e − 03)+	2.3546e − 01 (5.41e − 02)+	2.8084e − 01 (4.14e − 02)+	4.4670e − 01 (3.90e − 02)+	4.1291e − 01 (2.93e − 02)+
MPSO/DD	5.5034e − 02 (4.57e − 02)	**8.0679e-02 (5.53e-02)**	**1.3641e-01 (6.95e-02)**	**1.2149e-01 (6.36e-02)**	**1.1233e-01 (6.78e-02)**	**1.5372e-01 (7.04e-02)**
DTLZ6						
NSGA-III	**1.0920e-02 (2.02e-03)**−	1.9828e − 01 (8.23e − 02)+	2.3869e + 00 (9.62e − 01)+	2.9938e + 00 (1.78e + 00)+	3.5362e + 00 (1.21e + 00)+	2.4314e + 00 (1.78e + 00)+
KnEA	3.1273e − 03 (1.49e − 04)−	3.6151e − 01 (3.93e − 02)+	2.0387e + 00 (3.46e − 01)+	3.1093e + 00 (5.21e − 01)+	2.7357e + 00 (1.46e − 01)+	2.4779e + 00 (6.95e − 01)+
RVEA	7.5146e − 02 (1.48e − 02)+	1.7343e − 01 (2.77e − 02)+	3.1417e − 01 (8.17e − 02)+	2.1012e − 01 (4.45e − 02) ≈	2.1853e − 01 (7.15e − 02)+	3.9287e − 01 (2.01e − 01)+
MOEA/DD	2.5306e − 02 (4.02e − 04)−	**9.8550e-02 (8.81e-03)** ≈	3.8069e − 01 (1.81e − 01)+	2.2380e − 01 (4.98e − 02) ≈	1.9315e − 01 (2.58e − 03)+	2.8342e − 01 (1.22e − 01) ≈
SPEAR	3.4183e − 02 (4.53e − 03)−	5.9272e − 01 (3.05e − 01)+	2.9723e + 00 (3.67e − 01)+	8.6588e + 00 (3.59e − 01)+	9.3693e + 00 (2.72e − 01)+	9.2558e + 00 (2.11e − 01)+
GrEA	2.2218e − 02 (1.70e − 04)−	2.9301e − 01 (5.00e − 02)+	9.3493e − 01 (3.17e − 01)+	1.6623e + 00 (2.32e − 01)+	2.4890e + 00 (4.34e − 01)+	3.8828e + 00 (5.69e − 01)+
BiGE	6.2797e − 01 (6.78e − 02)+	6.8873e − 01 (6.57e − 02)+	6.9164e − 01 (1.32e − 01)+	8.4040e − 01 (1.29e − 01)+	9.5994e − 01 (3.40e − 01)+	8.8125e − 01 (2.49e − 01)+
MPSO/DD	6.0725e − 02 (2.55e − 02)	1.0601e − 01 (3.32e − 02)	**2.1684e-01 (1.43e-01)**	**2.0149e-01 (1.10e-01)**	**1.4478e-01 (1.07e-01)**	**2.6633e-01 (1.35e-01)**
DTLZ7						
NSGA-III	5.6349e − 02 (9.54e − 04)−	2.9277e − 01 (1.31e − 02)−	1.0225e + 00 (9.00e − 02)−	2.2502e + 00 (3.08e − 01)−	8.3893e + 00 (6.37e − 01)−	1.3650e + 01 (5.58e − 01)−
KnEA	**5.3322e-02 (2.77e-03)**−	2.3706e − 01 (4.97e − 03)−	8.5190e − 01 (1.49e − 01)−	**9.3770e-01 (3.25e-02)**−	5.2184e + 00 (2.36e + 00)−	2.0576e + 01 (2.88e + 00)+
RVEA	8.9720e − 02 (4.74e − 03)−	5.0272e − 01 (9.49e − 03)−	1.2736e + 00 (2.72e − 01)−	3.7764e + 00 (4.28e − 01)−	**5.1441e+00 (7.78e-01)**−	**5.4032e+00 (7.37e-01)**−
MOEA/DD	1.7328e − 01 (1.12e − 01)−	2.6250e + 00 (9.17e − 01)+	1.6053e + 00 (4.95e − 01)−	8.8923e + 00 (3.85e − 03)+	1.1465e + 01 (1.00e + 00)+	1.5919e + 01 (4.62e − 01)+
SPEAR	7.2926e − 02 (1.20e − 03)−	3.5525e − 01 (5.18e − 03)−	1.5034e + 00 (1.10e − 01)−	3.3083e + 00 (1.13e − 01)−	9.1617e + 00 (1.01e − 01)−	1.6694e + 01 (1.28e + 00)+
GrEA	7.4719e − 02 (2.75e − 03)−	**2.3139e-01 (9.21e-03)**−	**8.3820e-01 (2.73e-02)**−	2.8285e + 00 (9.88e − 02)−	1.0099e + 01 (9.01e − 01) ≈	1.7371e + 01 (1.45e + 00)+
BiGE	6.3840e − 02 (2.43e − 03)−	3.4243e − 01 (6.49e − 02)−	1.8332e + 00 (2.50e − 01)−	4.1036e + 00 (4.31e − 01)−	1.0714e + 01 (3.29e − 01) ≈	1.6136e + 01 (3.72e − 01)+
MPSO/DD	3.6729e − 01 (5.82e − 02)	1.1094e + 00 (1.14e − 01)	3.3395e + 00 (4.37e − 01)	4.5174e + 00 (5.10e − 01)	1.0304e + 01 (7.99e − 01)	1.4504e + 01 (5.38e − 01)

The best results are highlighted

Table 5 Performance of IGD result on DTLZ1–DTLZ7 test problems, where MPOS/DD is better than (+), worst than (−) and approximate to (≈) each of the seven compared algorithms according to the Wilcoxon rank sum test

MPOS/DD *v.s.*	NSGA-III	KnEA	RVEA	MOEA/DD	SPEAR	GrEA	BiGE	NMPSO
+	24/42	30/42	21/42	17/42	30/42	25/42	31/42	26/42
−	16/42	12/42	18/42	17/42	12/42	12/42	10/42	13/42
≈	2/42	0/42	3/42	8/42	0/42	5/42	1/42	3/42

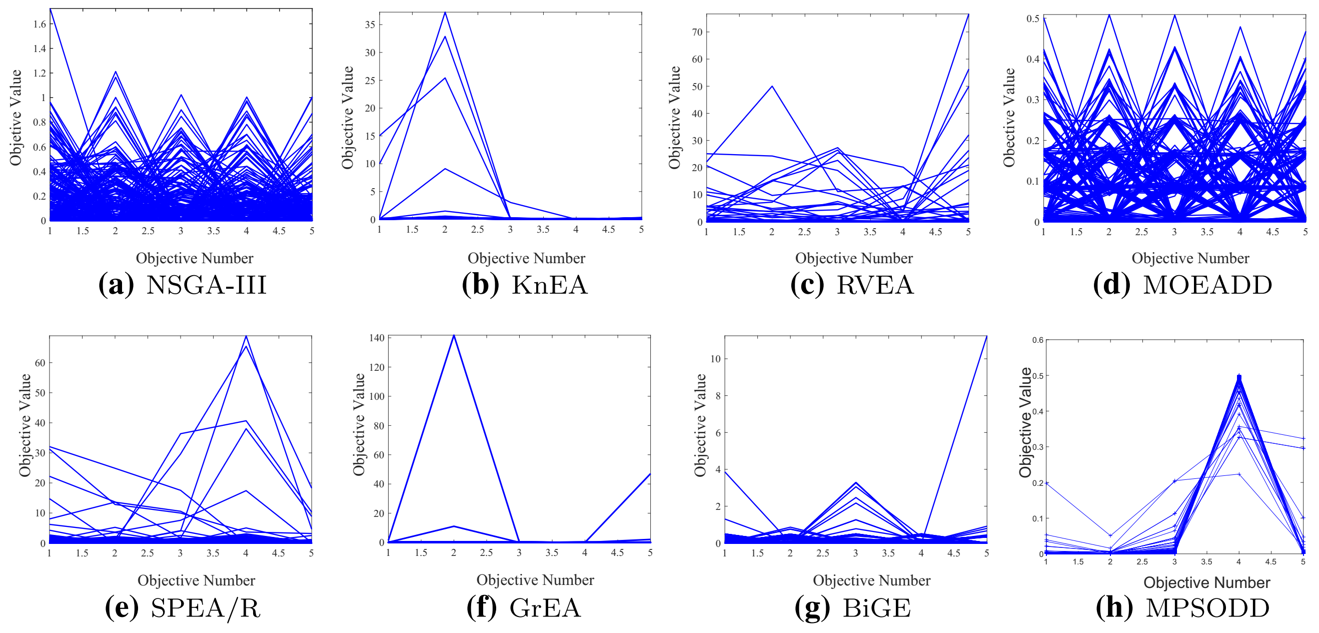

Fig. 3 Parallel coordinates of nondominated fronts obtained by eight algorithms on the five-objective DTLZ1 problem

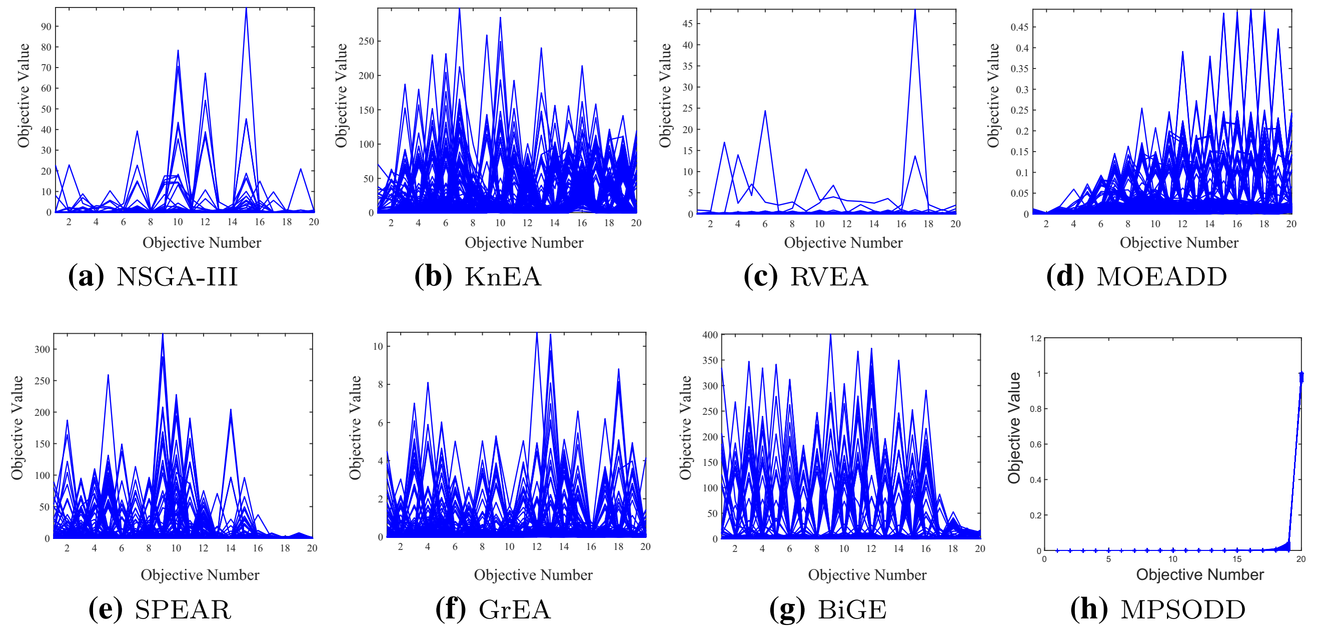

Fig. 4 Parallel coordinates of nondominated fronts obtained by 8 algorithms on the 20-objective DTLZ1 problem

Table 6 The statistical results(mean and standard deviation) of the IGD values obtained by NMPSO and MPSO/DD on DTLZ1 TO DTLZ7

M Problem	3	5	8	10	15	20
DTLZ1						
NMPSO	$2.8754e-01\ (4.17e-01) \approx$	**2.6906e-01 (1.94e-01)**$-$	**3.4336e-01 (3.08e-01)**$-$	$1.1376e+00\ (3.31e+00) \approx$	$3.5277e+01\ (9.96e+00)+$	$3.1479e+01\ (7.50e+00)+$
MPSO/DD	**2.3029e-01 (1.91e-01)**	$4.1405e-01\ (1.78e-01)$	$4.5166e-01\ (1.29e-01)$	**4.6162e-01 (1.89e-01)**	**6.1970e-01 (2.87e-01)**	**5.5316e-01 (1.84e-01)**
DTLZ2						
NMPSO	$6.4217e-02\ (2.45e-03)+$	$1.8862e-01\ (1.34e-03)+$	$3.5779e-01\ (1.47e-03)+$	$4.2514e-01\ (1.80e-03)+$	$7.8355e-01\ (5.36e-02)+$	$8.4976e-01\ (3.99e-02)+$
MPSO/DD	**4.1172e-02 (6.13e-05)**	**1.6113e-01 (3.21e-04)**	**3.2029e-01 (1.42e-03)**	**4.0232e-01 (1.83e-03)**	**6.0687e-01 (4.12e-02)**	**7.8543e-01 (4.84e-02)**
DTLZ3						
NMPSO	$2.9643e+01\ (1.43e+01)+$	$4.1312e+01\ (1.18e+01)+$	$4.2420e+01\ (3.48e+01)+$	$6.5367e+01\ (3.53e+01)+$	$1.6376e+02\ (6.80e+01)+$	$1.6161e+02\ (7.26e+01)+$
MPSO/DD	**4.8332e+00 (1.49e+00)**	**3.6665e+00 (1.45e+00)**	**3.2763e+00 (1.24e+00)**	**3.2172e+00 (1.91e+00)**	**3.0568e+00 (1.16e+00)**	**3.7659e+00 (2.33e+00)**
DTLZ4						
NMPSO	$1.1136e-01\ (1.47e-01)+$	$1.9018e-01\ (1.72e-03)+$	$3.8959e-01\ (5.20e-02)+$	**4.3746e-01 (1.13e-02)**$-$	$6.7780e-01\ (6.54e-02)+$	$6.2800e-01\ (1.56e-02) \approx$
MPSO/DD	**4.1262e-02 (6.58e-05)**	**1.6487e-01 (6.48e-04)**	**3.5425e-01 (5.54e-03)**	$4.5638e-01\ (9.80e-03)$	**6.2425e-01 (3.95e-03)**	**6.2293e-01 (3.72e-03)**
DTLZ5						
NMPSO	**1.3443e-02 (2.71e-03)**$-$	**4.4023e-02 (5.34e-03)**$-$	$7.1603e-01\ (1.63e-01)+$	$6.8691e-01\ (2.23e-01)+$	$7.5663e-01\ (1.78e-02)+$	$7.7782e-01\ (7.28e-02)+$
MPSO/DD	$5.5034e-02\ (4.57e-02)$	$8.0679e-02\ (5.53e-02)$	**1.3641e-01 (6.95e-02)**	**1.2149e-01 (6.36e-02)**	**1.1233e-01 (6.78e-02)**	**1.5372e-01 (7.04e-02)**
DTLZ6						
NMPSO	**1.2694e-02 (2.24e-03)**$-$	**3.7620e-02 (5.22e-03)**$-$	$7.0954e-01\ (1.34e-01)+$	$7.4209e-01\ (2.28e-16)+$	$7.4209e-01\ (2.28e-16)+$	$7.4803e-01\ (2.66e-02)+$
MPSO/DD	$6.0725e-02\ (2.55e-02)$	$1.0601e-01\ (3.32e-02)$	**2.1684e-01 (1.43e-01)**	**2.0149e-01 (1.10e-01)**	**1.4478e-01 (1.07e-01)**	**2.6633e-01 (1.35e-01)**
DTLZ7						
NMPSO	**5.2447e-02 (1.88e-03)**$-$	**2.2565e-01 (6.09e-03)**$-$	**8.6201e-01 (1.80e-01)**$-$	**9.4817e-01 (8.01e-02)**$-$	**2.7349e+00 (3.18e-01)**$-$	**2.5889e+00 (1.44e-01)**$-$
MPSO/DD	$3.6729e-01\ (5.82e-02)$	$1.1094e+00\ (1.14e-01)$	$3.3395e+00\ (4.37e-01)$	$4.5174e+00\ (5.10e-01)$	$1.0304e+01\ (7.99e-01)$	$1.4504e+01\ (5.38e-01)$

The best results are highlighted

20 objectives, which showed the competition of our proposed method to solve problems with high dimensions on objectives. Table 5 gives a summary on the results given in Table 4. From Table 5, we can clearly see that generally, MPSO/DD obtained better results than NSGA-III, KnEA, RVEA, SPEAR, GrEA, and BiGE, and competitive results with MOEA/DD.

To show the effectiveness of our proposed algorithm, Figs. 3 and 4 plot the parallel coordinates of the non-dominated solution set obtained by different algorithms on 5-objective and 20-objective DTLZ1 test problems, respectively. From Figs. 3 and 4, we can see that the objective values of MOEA/DD and MPSO/DD can decline much quicker than other algorithms, and the solutions are better distributed than other algorithms in a limited number of evaluations. While compared to MOEA/DD, we can see that the performance of MPSO/DD is comparative on 5-objective DTLZ1, but not better than MOEA/DD on 20-objective DTLZ1. The reason, we analyze, is that the diversity of MPSO/DD is still not better than that of MOEA/DD on DTLZ1. Therefore, to see whether our proposed MPSO/DD algorithm is competitive with those many-objective optimization algorithms based on PSO, we compare the results on DTLZ with NMPSO [17], in which a balanceable fitness estimation method and a novel velocity update equation were presented so as to effectively solve the many-objective optimization problems. Table 6 shows the mean IGD results of MPSO/DD compared to NMPSO, where the best results are highlighted. From Table 6, we can see that the MPSO/DD obtained better results than NMPSO on DTLZ1–DTLZ6 with high-dimensional objectives, which further showed that our proposed MPSO/DD is competitive to solve the problems with high-dimensional objectives. However, the results on DTLZ7 obtained by MPSO/DD is not better than NMPSO, the reason, we analyze, may result from the BFE method proposed in NMPSO which strongly prefers the solutions with well converged and less crowded.

Table 5 also shows the summary on the results given in Table 6. From Table 5, we can clearly see that the proposed MPSO/DD obtained 26/42 better results than NMPSO, which showed the better performance of our proposed MPSO/DD than NMPSO.

Conclusion

This paper proposed a modified particle swarm optimization algorithm, in which the decomposition strategy and different ideal points are utilized, for many-objective problems. The experimental results showed that the proposed algorithm has advantages on solving problems with high-dimensional objective space, but many works are still remained for us to study further. In the future, we will try to add some strategies to prevent the algorithm from falling into local optima to achieve better results on all problems.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant no.: 61876123), Natural Science Foundation of Shanxi Province (201801D121131, 201901D111264, 201901D111262), Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province, Shanxi Science and Technology Innovation project for Excellent Talents (201805D211028), Postgraduate Education Innovation Project of Shanxi Province (2019SY494), the Doctoral Scientific Research Foundation of Taiyuan University of Science and Technology (20162029), and the China Scholarship Council (CSC).

References

1. Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76
2. Beume N, Naujoks B, Emmerich M (2007) Sms-emoa: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
3. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. Inf Sci 291(6):43–60
4. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791
5. Clerc M, Kennedy J (2002) The particle pwarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
6. Coello CAC, Sierra MR (2004) A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Mexican international conference on artificial intelligence
7. Deb K (2001) Multiobjective optimization using evolutionary algorithms. Springer, New York
8. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601
9. Deb K, Pratap A, Agarwal S, Meyarivan T, Fast A (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
10. Herrero JG, Berlanga A, López JMM (2009) Effective evolutionary algorithms for many-specifications attainment: application to air traffic control tracking filters. IEEE Trans Evol Comput 13(1):151–168
11. Jiang S, Yang S (2017) A strength pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization. IEEE Trans Evol Comput 21(3):329–346

12. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multiobjective optimization. Evol Comput 10(3):263–282
13. Li K, Deb K, Zhang Q, Kwong S (2015) An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans Evol Comput 19(5):694–716
14. Li K, Kwong S, Zhang Q, Deb K (2015) Interrelationship-based selection for decomposition multiobjective optimization. IEEE Trans Cybern 45(10):2076–2088
15. Li K, Zhang Q, Kwong S, Li M, Wang R (2014) Stable matching based selection in evolutionary multiobjective optimization. IEEE Trans Evol Comput 18(6):909–923
16. Li M, Yang S, Liu X (2015) Bi-goal evolution for many-objective optimizationproblems. Artif Intell 228(C):45–65
17. Lin Q, Liu S, Zhu Q, Tang C, Song R, Chen J, Coello CAC, Wong KC, Zhang J (2016) Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. IEEE Trans Evol Comput 22(1):32–46
18. Liu HL, Gu F, Zhang Q (2014) Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. IEEE Trans Evol Comput 18(3):450–455
19. Mkaouer MW, Kessentini M, Bechikh S, Deb K, Ó Cinnéide M (2014) High dimensional search-based software engineering: finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III. In: Proceedings of the 2014 annual conference on genetic and evolutionary computation, ACM, pp 1263–1270
20. Qin S, Sun C, Jin Y, Lan L, Tan Y (2019) A New Selection Strategy for Decomposition-based Evolutionary Many-Objective Optimization. In: 2019 IEEE congress on evolutionary computation (CEC), IEEE, pp 2426–2433
21. Sülflow A, Drechsler N, Drechsler R (2007) Robust multi-objective optimization in high dimensional spaces. In: International conference on evolutionary multi-criterion optimization, Springer, New York, pp 715–726
22. Tian Y, Cheng R, Zhang X, Cheng F, Jin Y (2017) An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility. IEEE Trans Evol Comput PP(99):1–1
23. Wang G, Jiang H (2007) Fuzzy-dominance and its application in evolutionary many objective optimization. In: International conference on computational intelligence security workshops
24. Wang H, Jiao L, Yao X (2015) Two_arch2: An improved two-archive algorithm for many-objective optimization. IEEE Trans Evol Comput 19(4):524–541
25. Wang Z, Zhang Q, Li H (2015) Balancing convergence and diversity by using two different reproduction operators in moea/d: some preliminary work. In: 2015 IEEE international conference on systems, man, and cybernetics, pp 2849–2854. https://doi.org/10.1109/SMC.2015.496
26. Xiang Y, Zhou Y, Li M, Chen Z (2017) A vector angle based evolutionary algorithm for unconstrained many-objective optimization. IEEE Trans Evol Comput 21(1):131–152
27. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736
28. Ye T, Ran C, Zhang X, Jin Y (2017) Platemo: a matlab platform for evolutionary multi-objective optimization. IEEE Comput Intell Magn 12(4):73–87
29. Ying S, Li L, Zheng W, Li W, Wang W (2017) An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity. Appl Soft Comput 57:S1568494617301655
30. Yuan Y, Hua X, Bo W, Xin Y (2016) A new dominance relation based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(1):16–37
31. Zhang Q, Hui L (2007) Moea/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
32. Zhang X, Tian Y, Jin Y (2015) A knee point driven evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 19(6):761–776
33. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: International conference on parallel problem solving from nature, Springer, New York, pp 832–842