Two-stage multi-tasking transform framework for large-scale many-objective optimization problems

Lu Chen¹ · Handing Wang¹ · Wenping Ma¹

Received: 3 September 2020 / Accepted: 13 January 2021 / Published online: 17 February 2021
© The Author(s) 2021

Abstract

Real-world optimization applications in complex systems always contain multiple factors to be optimized, which can be formulated as multi-objective optimization problems. These problems have been solved by many evolutionary algorithms like MOEA/D, NSGA-III, and KnEA. However, when the numbers of decision variables and objectives increase, the computation costs of those mentioned algorithms will be unaffordable. To reduce such high computation cost on large-scale many-objective optimization problems, we proposed a two-stage framework. The first stage of the proposed algorithm combines with a multi-tasking optimization strategy and a bi-directional search strategy, where the original problem is reformulated as a multi-tasking optimization problem in the decision space to enhance the convergence. To improve the diversity, in the second stage, the proposed algorithm applies multi-tasking optimization to a number of sub-problems based on reference points in the objective space. In this paper, to show the effectiveness of the proposed algorithm, we test the algorithm on the DTLZ and LSMOP problems and compare it with existing algorithms, and it outperforms other compared algorithms in most cases and shows disadvantage on both convergence and diversity.

Keywords Large-scale optimization · Many-objective optimization · Evolutionary algorithms · Multi-tasking optimization

Introduction

Conventional methods based on gradients cannot solve non-analytic optimization problems. Researchers in the past 2 decades have confirmed that evolutionary algorithms (EAs) are effective on black-box optimization problems. Also, the need to simultaneously optimize multiple indicators is very common for engineering optimization, which can be modeled as the multi-objective optimization problems (MOPs)

with M objectives as follows:

$$\begin{aligned} \min F(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s.t. } \mathbf{x} &\in X. \end{aligned} \quad (1)$$

Here, X is the feasible region of D decision variables ($\mathbf{x} = (x_1, \dots, x_D)$) [1].

Solving MOPs has two difficulties. First, traditional mathematical methods cannot solve MOPs by gradients. Second, the optimal solutions of an MOP is a solution set due to the conflict of multiple objective functions. The Pareto dominance relationship is a classical and widely used method for multi-objective optimization. It can be explained in this way, for any two solutions of an MOP illustrated by Eq. (1): solution S_A is known to Pareto dominate solution S_B , if and only if $f_i(S_A) \leq f_i(S_B)$ ($\forall i \in 1, 2, \dots, M$) and there exists at least one objective f_j ($j \in 1, 2, \dots, M$) satisfying $f_j(S_A) < f_j(S_B)$. The set of all the Pareto optimal solutions in the decision space is called the Pareto set (PS), and the projection of PS in the objective space is called the Pareto front (PF).

So far, multi-objective evolutionary algorithms (MOEAs) have been a successful tool for solving MOPs [2], because

This work was supported in part by the National Natural Science Foundation of China (no. 61976165).

✉ Handing Wang
hdwang@xidian.edu.cn

Lu Chen
lunulunulu@gmail.com

Wenping Ma
wpma@mail.xidian.edu.cn

¹ School of Artificial Intelligence, Xidian University, Xi'an 710071, Shaanxi, China

they can output the Pareto set in a single run. MOEAs using the Pareto dominance for their selection are popular. Many Pareto-based MOEAs have their own selection methodology. For example, NSGA-II [3] is based on a fast non-dominated sorting approach for selecting good solutions, and SPEA2 [4] is based on the non-dominated fitness allocation. Aggregation-based algorithms, such as MOEA/D [5], transform an MOP into a number of single-objective optimization problems and optimize those sub-problems simultaneously. Furthermore, the performance measurement can be employed in the selection process of MOEAs: IBEA [6], SMS-EMOA [7], and HypE [8] define their optimization goals in terms of different performance indicators.

As the number of objectives increases, MOPs become many-objective optimization problems (MaOPs), the ability of filtering out good individuals by the Pareto dominance degenerates. To improve the poor performance of the Pareto dominance, many algorithms combine the Pareto dominance-based criterion with additional convergence-related metrics. In those methods, solutions are selected first based on the Pareto dominance, and then based on the convergence-related metric. Those representative approaches include grid-based evolutionary algorithm (GrEA) [9], preference-inspired co-evolutionary algorithm (PICEA-g) [10], knee point driven evolutionary algorithm (KnEA) [11], and many-objective evolutionary algorithm based on directional diversity and favorable convergence (MaOEA-DDFC) [12].

The large number of decision variables of MOPs (termed LSMOPs) pose challenges to existing MOEAs, because the searching space is high-dimensional, which exponentially increases the computation cost. The most straightforward way to deal with LSMOPs is to use the high-dimensional search, like CMOPSO [13,14] uses competitive particle swarm optimizer [15], which is good at high-dimensional optimization problems. To decrease the searching space, grouping strategies have been widely used in cooperative co-evolutionary (CC) framework for large-scale optimization. In CC framework, decision variables are decomposed into groups as different sub-problems for cooperative co-evaluation strategies. However, most CC algorithms are applied to single-objective optimization, except a few algorithms for MOPs [16]. In fact, the detected information of LSMOPs can further assist the grouping strategies. For example, DRMOS [17,18], MOEA/DVA [19], LMEA [20], and S3CMAES [21] divide the decision variables into different categories for applying different searching strategies. In addition, the problem transformation method is an alternative way to reduce the high complexity of LSMOPs. For example, LSMOF [22] reduces the number of decision variables greatly by transforming the original LSMOP into a number of weight optimization sub-problems. WOF [23] is another problem transformation method combining with a grouping strategy.

As mentioned above, both the high-dimensional decision and objective space of MOPs increase the hardness and computation cost of solving large-scale MaOPs by existing MOEAs which are specially designed for either MaOPs or large-scale MOPs. To address both issue, we borrow the parallelism of multifactorial optimization (MFO) [24], where a single population is employed to optimize multiple optimization problems simultaneously, for solving large-scale many-objective optimization problems (LSMaOPs). Our proposed algorithm is an two-stage framework combined with the multi-tasking optimization (termed TSMTF).

The rest structure of this paper is as follows: the next section will introduce the related works of LSMaOPs and MFO idea. The detailed the proposed algorithm TSMTF will be shown in the continuous section. Then, the experimental results are presented in the consequent section. Finally, the last section makes the conclusion.

Related work

Large-scale multi-objective optimization

To address the problems with a large number of decision variables, some representative MOEAs (CMOPSO [13], [14], S3CMAES [21], MOEA/DVA [19], and LMEA [20], WOF [23]) to reduce the large search space have been proposed. We will discuss them in detail.

CMOPSO [13] employs a competitive swarm optimizer [15]. With the good search ability for high-dimensional space of competitive swarm optimizer, CMOPSO outperforms other MOEAs, but it cannot solve large-scale MOPs. Therefore, it is necessary to further reduce the decision space.

The algorithm in [14] adopts a two-stage framework, unlike the normal particle swarm optimization algorithms focusing on updating velocity, it proposes a new updating strategy to update the particles' positions with a new competitive mechanism, first pre-updates the particles' positions by its previous velocity and then learn from the leader's position. The experiment results show its position updating strategy is effective on LSMOPs.

MOEA/DVA [19], LMEA [20], and S3CMAES [21] are three examples of decision variable analysis-based algorithms. MOEA/DVA [19] solves problems by a clustering method based on the dominance-based relationship, it divides the decision variables into the following three groups:

- Convergence-related variables: the decision variables contribute to convergence.
- Diversity-related variables: the decision variables contribute to diversity.
- Mixed variables: the decision variables contribute to both convergence and diversity.

With those groups, MOEA/DVA re-structures the problem into a set of sub-problems. Then, the convergence-related variables are optimized firstly to get candidate solutions which are close to the PF. Then, MOEA/DVA treats both diversity-related and mixed variables as diversity-related variables to enhance the diversity.

In LMEA, the decision variables are divided into two classes based on the disturbance test on each decision variable: convergence- and diversity-related decision variables. Then, both classes of decision variables are optimized to improve convergence and diversity separately within a population. It also employs a fast non-dominated sorting method [25] to further decrease the computation cost.

S3CMAES [21] divides the decision variables into diversity-related and convergence-related groups by a clustering strategy. After that, the convergence-related variables are divided into sub-groups and each sub-group is optimized independently using covariance matrix adaptation evolution strategy (CMA-ES). S3CMAES shows good performance on LSMOPs. However, CMA-ES is repeatedly employed for each sub-group, which makes its cost high.

However, the decision variable clustering methods in above three MOEAs require a large number of function evaluations, which makes them hard to be used for the real-world applications.

Furthermore, problem transformation methods are an alternative way to deal with LSMOPs. In both [26] and [22], the algorithms change the original problem by introducing a set of weight variables in the decision space. The experimental results show its effectiveness on convergence. For example, a solution $\mathbf{X} = (x_1, x_2, \dots, x_D)$ for the problem $f(\mathbf{X})$ can be transformed to sub-problems $f(\phi(\omega, \mathbf{X}))$, where $\phi(\omega, \mathbf{X})$ can be defined as below:

$$\phi(\omega, \mathbf{X}) = (\omega_1 \chi_1, \omega_1 \chi_2, \dots, \omega_1 \chi_D, \dots, \omega_k \chi_{D-k+1}, \omega_k \chi_{D-k+2}, \dots, \omega_k \chi_D). \quad (2)$$

Thus, the problem with D decision variables is transformed into several k -dimensional problems. Thanks to the reduction of the decision space, the searching cost has been significantly reduced. However, such transformation can be viewed as a lossy dimension reduction, which causes stability issues. Specially in LSMOF [22], its transformed objective function is based on the hypervolume (HV) indicator, which is computationally expensive for MaOPs. Therefore, LSMOF cannot solve LSMaOPs.

By combining grouping method and problem transformation method, WOF [23] also shows good performance on LSMOPs, it divides the decision variables into groups with its grouping strategies, and each group is assigned with a weight. Unlike other grouping-based algorithms, it cannot optimize the sub-populations separately, WOF transforms the original LSMOP into weight optimization problems and optimize the

transformed problems with a whole population as LSMOF [22].

Multi-tasking optimization

MFEA [24] is a parallel method to solve multiple problems at the same time. It defines the following properties for each individual p_i in the population P :

- Factorial cost: The factorial cost of p_i on task problem j is T_i^j .
- Factorial rank: In the population, each individual is assigned a rank for each task; for example, in terms of task j , p_i 's rank is R_i^j .
- Scalar fitness: Each individual is assigned the fitness φ_i based its best-performing task.
- Skill factor: Each individual is marked its best-performing task by $\tau_i = \text{argmin}_j r_i^j$ where $j \in 1, \dots, M$.

Algorithm 1: Main framework of MFEA

Input: T (tasks), N (Population size), E (total number of evaluations)

Output: The final population

```

1  $P \leftarrow \text{Initialization}(T, P)$ 
2 while  $\text{evaluations} < E$  do
3   Calculate the task rank on each task, and Scalar Fitness, skill
   factor for each individual  $P_i$ .
4    $S \leftarrow$  Offsprings generated by Algorithm 2.
5    $P \leftarrow$  Select the good individuals based on scalar fitness  $\varphi$ .
6 end
```

Thus, multiple tasks can be turned into a one-objective problem that all the problems use the φ of each individual as the selection criterion. The selection of this algorithm is based on the scalar fitness. In other words, the algorithm concerns more about the best individual in each task. The offspring generation is also based on the skill factor. The parents P_a and P_b are randomly chosen from the population, if their skill factors are different, they will execute crossover operation in a low probability; otherwise, they will execute the crossover or mutation operations. More details of MFEA can be found in Algorithms 1 and 2.

Proposed algorithm

Framework

To address the issue of LSMOF [22] on LSMaOPs, we borrow the main idea of KnEA [11] which optimizes the distance from knee points to the hyperplane to replace the HV calculations of LSMOF in the proposed algorithm. To further reduce

Algorithm 2: Offspring generation in MFEA

Input: T (tasks), P (Parent Population), mt (the probability of variation)

Output: The Offsprings

```

1  $P_a$  and  $P_b$  are randomly chosen from the population, and
  generate a random number  $q_1$ 
2 if  $P_a$  and  $P_b$  have the same skill factor and  $q_1 > mt$  then
3    $O_c, O_d \leftarrow$  Execute crossover  $P_a$  and  $P_b$  each of them
  generate a random number  $q_2$ 
4   if  $q_2 < mt$  then
5     It only calculates the  $T_i$  corresponding to  $p_a$ 's skill factor
      $i$ , the fitness of the rest tasks are set to infinity.
6   end
7   else
8     It only calculates the  $T_j$  corresponding to  $p_b$ 's skill factor
      $j$ , the fitness of rest tasks are set to infinity.
9   end
10 end
11 else
12    $O_c \leftarrow$  apply mutation to  $P_a$ 
13   It only calculates the  $T_i$  corresponding to  $p_a$ 's skill factor  $i$ ,
  the fitness of the rest tasks are set to infinity.
14    $O_d \leftarrow$  apply mutation to  $P_b$ 
15   It only calculates the  $T_j$  corresponding to  $p_b$ 's skill factor  $j$ ,
  the fitness of the rest tasks are set to infinity.
16 end

```

the required number of function evaluations, we employ MFEA in TSMTE.

The structure of the proposed algorithm contains two main stages, which is highlighted in Algorithm 3. The first stage aims to find out the solutions with good convergence; the second stage focuses on the diversity improvement; then, the population with good convergence and diversity is embedded in NSGA-III [27]. When the computation budget runs out, the proposed algorithm outputs the obtained optimal solutions.

Algorithm 3: Main framework of algorithm

Input: T (original problem), N (population size), R_1 (number of reference points in the first stage), R_2 (number of reference points in the second stage), NW (the number of the weights), NE (number of evaluations), tr_1 (percentage of cost for the first stage), tr_2 (percentage of cost for the second stage)

Output: The final population

```

1  $P \leftarrow$  Initialization( $P$ )
2 while  $tr < tr_1 \times NE$  do                                     # stage1
3    $T_1 \leftarrow$  Problem reformulation ( $(P, R_1, T)$ )
4    $W_1 \leftarrow$  Optimal weights obtained by multi-tasking
  optimizing ( $T_1$ )
5    $P_1 \leftarrow$  Transformed  $W_1$  in the decision space.
6 end
7 while  $tr < tr_2 \times NE$  do                                     # stage2
8   Choose  $R_2$  points in the population  $P_2 \leftarrow R_2$  points after
  diversity improvement.
9 end
10  $P \leftarrow$  Embedded NSGA-III( $(P_2, N, T)$ )

```

Stage 1: Bi-directional weight variable associated multi-tasking strategy

This stage aims to find the better individuals by reducing the dimension of the searching space. One individual A in the decision space in Fig. 1 can be expressed as $A_o = (x_1, \dots, x_n)$. Then, two directions from the extreme points of the decision space can be generated as: $A_u = O + A_o$, $A_d = T - A_o$. Then, we search best solutions in both directions by optimizing the weights λ in two sub-problems, as shown in Fig. 2.

In this stage, the proposed algorithm chooses $2R1$ directions which are from $R1$ diverse individuals on the current PF. Since HV is very computationally expensive for MaOPs, it cannot be employed as the objective function of those $2R1$ sub-problems. In this work, we set the distance to the estimated ideal point as the objective function of those $2R1$ sub-problems.

To simultaneously find the optimal λ of each sub-problem, we employ MFEA as the optimizer. In this method, after the normalization, the directions can be expressed as $D = (d_1, d_2, \dots, d_{2R1})$. We use a weight vector $V_0 = (\lambda_1, \lambda_2, \dots, \lambda_{2R1})$ to represent solutions $P_1, P_2, \dots, P_{2R1}$, by $V_0 \cdot D$.

We consider each sub-problem as one task: P_i is generated by directions d_i and λ_i ($V_0 \cdot D$) for optimizing the i th task, which can be seen as an instance of the task T_i . Thus, $P_1, P_2, \dots, P_{2R1}$ corresponding to $T_1, T_2, \dots, T_{2R1}$'s instance. For each instance, there is a function to evaluate its performance, as MFEA, which is called factorial cost. The function is the distance of P_i to the estimated ideal point. In this case, the weight vectors are taken for the individuals in MFEA need to be optimized. After sorting the obtained factorial costs, MFEA solves $2R1$ sub-problems efficiently, as shown in Algorithm 4.

Algorithm 4: Multi-tasking weight optimization

Input: D (directions), V (weight vectors), TM (Termination computation times)

```

1  $V \cdot D$  generate a group of individuals, for each  $V_i$ , contains its
  scalar fitness and skill factor  $K_i$ 
2 while  $evaluation < TM$  do
3   Generate the offspring weight vectors( $C$ )(refer to Algorithm
  3), and  $C \cdot D$  are the individuals.
4   Evaluate the individuals for selected optimization tasks only.
5    $V = V \cup C$ .
6   Update the scalar fitness and the skill factors of each weight
  vector select the good vectors.
7 end
Output: vectors, individuals

```

Fig. 1 Illustration of the chosen points A, B for generating directions

Fig. 2 Illustration of the individuals generated by the weight vector and directions

Stage 2: Diversity improvement with multi-tasking strategy

This stage aims to improve the diversity. In this stage, first, we choose $R2$ individuals as the reference points; one of them is the shortest distance to the obtained ideal point and the rest are chosen from the most dispersed individuals. As NSGA-III, each solution is associated with its nearest reference point, the most dispersed individuals are those reference points who has the fewest associate points.

Second, we expect the population evolves to those different reference points to improve diversity. In this case, we use the PBI value [5] as the criterion to evaluate the similarity to each reference point in the objective space. To push the population to those reference points, we consider each PBI distance to each direction as a task. For example, Taking $R2$ reference individuals: $L(l_1, l_2, \dots, l_{R2})$ as an example, the multi-tasking population P is to minimize $PBI(P, l_i)$, $i \in (1, \dots, R2)$ simultaneously; each individual's factorial cost is the set of PBI values to those $R2$ reference individuals. Similar to the first stage, we employ MFEA to optimize those $R2$ sub-problems.

Experiments and discussion

Parameter settings

To test the performance of the proposed algorithm, the DTLZ [28] and LSMOP [29] problems are chosen as the test problems, because their numbers of the decision variables and objective functions are scalable. In the experiment, the decision variables are set to 100D, 200D, and 500D, while the numbers of objectives are set to 3, 5, and 10. To prove the effectiveness of the proposed algorithm in this paper, several existing popular algorithms are employed as the compared algorithm: MOEA/D [5], NSGA-II [3], and NSGA-III [27]. In addition, we choose some other algorithms for MaOPs and large-scale MOPs: KnEA [11] and LMEA [20]. For DTLZ problems, in addition to those algorithms mentioned in this paragraph, we also choose S3CMAES and CMOPSO for the comparison. The settings of those algorithms are recommended values as their original papers. The source codes of the compared algorithms are from PlatEMO [30].

For all the experiments, each algorithm runs 30 times independently for each test problem. In this paper, we use IGD [5] as the indicator to assess the performance of compared algorithms, which has been widely used as a performance indicator. In short, the smaller the IGD is, the better the algorithm performs.

The parameters of the proposed framework are set as follows. For DTLZ problems, the population size for the 3-objective and the 5-objective problems are set to 120 and

150, while that for the 10-objective problems is 200, and the corresponding stopping criteria are 30,000, 50,000, and 50,000 function evaluations. In the 200D decision variables' situation, the population are set to 200, 300, and 300, with 50,000, 60,000, and 80,000 function evaluations for 3-, 5-, and 10-objective problems respectively. And in the 500D decision variable space, the population are set to 300, 500, and 500, with 100,000, 200,000, and 200,000 function evaluations for 3-, 5-, and 10-objective problems, respectively. As for LSMOPs, for 3-objective problems, the population size is set to 200 and the stopping criteria are set to 50000 function evaluations, for 5-objective problems, the population size is 200, and the evaluation times are set to 100,000, and for 10-objective problems, the size of the population is set to 200, the evaluation times are set to 100,000. The stage 1 uses 60% function evaluations, and the direction size is 10. For the stage 2, the number of the reference points is set to 11. About the specific method of crossover of TSMTE, we adopt the simulated binary crossover (SBX) [31], as for the mutation, we adopt the polynomial mutation [32]. The mutation probability stays at $1/D$ where D is the number of decision variables and the crossover probability keeps 1, both of the distribution index is set as recommended in [31].

Effects of stage 1

In this subsection, we discuss the effect of the first stage on the proposed algorithm. For the stage 1, we aim to improve the convergence of the population with a limited number of function evaluations. In this stage, we make use of the bi-directional method to search the individuals, which combines with MFEA to improve the efficiency. We compare TSMTE with the version without MFEA in the stage 1 (termed TSMTE1) on the 3-, 5-, and 10-objective DTLZ problems with 100, 200, and 500 decision variables. In TSMTE1, the weights are optimized serially, it optimizes directions one by one. For each direction, it searches for the best parameters λ instead of weight vector $V = (\lambda_1, \lambda_2, \dots, \lambda_H)$.

As shown in Tables 1, 2 and 3, TSMTE behaves better than TSMTE1 in most cases. In general, the framework embedded with MFEA has tiny advantages in terms of the IGD values on DTLZ1 and DTLZ3. For the other problems, their difference is easy to be distinguished. As the dimension of the decision variable increases, the overall advantage of TSMTE can be kept. Similarly, with the number of the decision variables increase and the number of the objectives remains unchanged, the experimental results do not show any changing trends. The main reason of this phenomena is that MFEA in this stage can reduce the number of function evaluations. In other words, the proposed algorithm is efficient.

Table 1 IGD Results of TSMTF and TSMTF without MFEA in stage 1 (TSMTF1) on DTLZ problems with 100 decision variables

Problem	M	D	TSMTF_1	TSMTF
DTLZ1	3	100	2.3159e−1 (2.65e−1) ≈	1.9095e−1 (1.66e−1)
DTLZ1	5	100	2.7756e−1 (2.10e−1) ≈	2.6679e−1 (2.19e−1)
DTLZ1	10	100	3.9876e−1 (3.86e−1) ≈	2.6186e−1 (9.23e−2)
DTLZ2	3	100	2.1460e−1 (2.42e−2) −	1.1166e−1 (2.11e−2)
DTLZ2	5	100	4.5465e−1 (4.07e−2) −	3.2963e−1 (3.04e−2)
DTLZ2	10	100	9.8898e−1 (8.28e−2) −	9.2663e−1 (9.32e−2)
DTLZ3	3	100	9.8721e−1 (8.97e−1) ≈	6.7821e−1 (4.68e−1)
DTLZ3	5	100	1.6214e+0 (2.26e+0) ≈	8.0925e−1 (4.14e−1)
DTLZ3	10	100	1.5312e+0 (1.41e+0) ≈	1.0043e+0 (2.66e−1)
DTLZ4	3	100	1.0179e+0 (1.60e−2) −	9.6252e−1 (2.94e−3)
DTLZ4	5	100	1.2219e+0 (2.17e−2) −	1.1332e+0 (5.09e−3)
DTLZ4	10	100	1.1463e+0 (1.14e−1) −	1.0204e+0 (8.40e−2)
DTLZ5	3	100	1.3201e−1 (2.28e−2) −	4.1641e−2 (8.07e−3)
DTLZ5	5	100	1.8208e−1 (3.56e−2) −	5.1708e−2 (1.44e−2)
DTLZ5	10	100	1.1096e−1 (1.95e−2) −	2.3729e−2 (9.69e−3)
DTLZ6	3	100	1.4503e+1 (3.55e+0) ≈	1.3122e+1 (2.58e+0)
DTLZ6	5	100	3.0875e+1 (3.57e+0) −	2.8037e+1 (3.26e+0)
DTLZ6	10	100	4.4653e+1 (2.68e+0) −	4.1285e+1 (4.85e+0)
/+/-/≈			0/11/7	

Table 2 IGD results of TSMTF and TSMTF without MFEA in stage 1 (TSMTF1) on DTLZ problems with 200 decision variables

Problem	M	D	TSMTF_1	TSMTF
DTLZ1	3	200	6.1334e−1 (9.45e−1) ≈	3.9263e−1 (5.25e−1)
DTLZ1	5	200	7.0291e−1 (1.28e+0) ≈	3.9704e−1 (4.02e−1)
DTLZ1	10	200	4.2077e−1 (3.14e−1) ≈	3.9067e−1 (4.12e−1)
DTLZ2	3	200	2.9387e−1 (4.74e−2) −	1.5868e−1 (2.71e−2)
DTLZ2	5	200	5.2812e−1 (3.25e−2) −	3.2017e−1 (2.38e−2)
DTLZ2	10	200	7.5416e−1 (3.62e−2) ≈	6.9526e−1 (3.32e−2)
DTLZ3	3	200	1.8777e+0 (2.02e+0) ≈	1.0958e+0 (1.06e+0)
DTLZ3	5	200	2.8784e+0 (3.23e+0) −	9.2110e−1 (1.28e+0)
DTLZ3	10	200	1.1404e+0 (5.36e−1) −	9.9869e−1 (2.49e−1)
DTLZ4	3	200	1.0861e+0 (2.73e−2) −	9.8995e−1 (1.09e−2)
DTLZ4	5	200	1.2615e+0 (6.13e−2) −	1.1444e+0 (8.89e−3)
DTLZ4	10	200	9.1953e−1 (8.10e−2) ≈	8.9596e−1 (8.36e−2)
DTLZ5	3	200	2.5744e−1 (5.29e−2) −	1.1647e−1 (1.83e−2)
DTLZ5	5	200	3.0727e−1 (2.37e−2) −	7.2253e−2 (1.27e−2)
DTLZ5	10	200	1.3487e−1 (1.47e−2) −	3.3920e−2 (5.52e−3)
DTLZ6	3	200	5.4213e+1 (8.10e+0) ≈	5.2788e+1 (6.14e+0)
DTLZ6	5	200	6.6262e+1 (6.70e+0) ≈	6.4504e+1 (5.79e+0)
DTLZ6	10	200	7.9500e+1 (8.46e+0) ≈	8.0260e+1 (8.06e+0)
/+/-/≈			0/9/9	

Table 3 IGD Results of TSMTF and TSMTF without MFEA in stage 1 (TSMTF1) on DTLZ problems with 500 decision variables

Problem	M	D	TSMTF_1	TSMTF
DTLZ1	3	500	9.8536e−1 (9.74e−1) −	2.9675e−1 (3.33e−1)
DTLZ1	5	500	7.6155e−1 (1.58e+0) ≈	2.1903e−1 (1.91e−1)
DTLZ1	10	500	4.9105e−1 (6.11e−1) ≈	2.8760e−1 (3.03e−1)
DTLZ2	3	500	4.3140e−1 (4.44e−2) −	2.0839e−1 (3.49e−2)
DTLZ2	5	500	2.7434e−1 (1.56e−2) −	2.1777e−1 (1.80e−2)
DTLZ2	10	500	1.0499e+0 (1.08e−1) −	9.9848e−1 (1.28e−1)
DTLZ3	3	500	7.2778e+0 (1.46e+1) ≈	1.9276e+0 (2.46e+0)
DTLZ3	5	500	8.0920e−1 (7.77e−1) ≈	1.0959e+0 (2.48e+0)
DTLZ3	10	500	1.3802e+0 (1.41e+0) ≈	1.0554e+0 (6.02e−1)
DTLZ4	3	500	1.1104e+0 (2.78e−2) −	1.0291e+0 (2.42e−2)
DTLZ4	5	500	1.1702e+0 (7.98e−3) −	1.1496e+0 (1.16e−2)
DTLZ4	10	500	1.2763e+0 (7.99e−2) +	1.4155e+0 (1.35e−1)
DTLZ5	3	500	4.2308e−1 (5.37e−2) −	1.7894e−1 (1.79e−2)
DTLZ5	5	500	1.9486e−1 (1.88e−2) −	7.5147e−2 (9.83e−3)
DTLZ5	10	500	1.5430e−1 (1.60e−2) −	6.5108e−2 (1.24e−2)
DTLZ6	3	500	1.4408e+2 (2.03e+1) ≈	1.3708e+2 (2.04e+1)
DTLZ6	5	500	1.6333e+2 (1.32e+1) −	1.5003e+2 (1.53e+1)
DTLZ6	10	500	2.0176e+2 (2.63e+1) ≈	1.9363e+2 (3.09e+1)
+/-/≈			1/10/7	

Comparative experiments

In this section, we compare the proposed algorithm with MOEA/D, NSGA-II, NSAG-III, KnEA, LMEA, S3CMAES, and CMOPSO on the large-scale DTLZ problems. To make the comparison clear, we make use of the Wilcoxon rank sum test [33], where TSMTF is the control algorithm. The symbols “+”, “−”, and “≈” correspond to the results of “better”, “worse”, and “nearly” compared with TSMTF.

As shown in Tables 4, 5 and 6, in terms of the overall results, it indicates that the proposed framework TSMTF is convergent and performs markedly better than the other algorithms on DTLZ1 and DTLZ3. On DTLZ2 and DTLZ4, and TSMTF also shows good convergence, but it does not show good performance compared with KnEA. We speculate that such situation might be caused by the characteristic of DTLZ2 and DTLZ4. Relatively speaking, LMEA has no advantage in this experiment because of the large amount of calculation on decision variables' classifications, and the number of calculations in this experiment may be too small to show better performance. S3CMAES is in the same situation. Its computation cost is too high to search each sub-population within a limited computation budget, and its clustering strategy also requires a large number of evaluations. The performance of CMOPSO is similar to the other compared algorithms, and it performs better than TSMTF on DTLZ2 and DTLZ4 occasionally. MOEA/D and NSGA-III perform best among all the algorithms, while the other algorithms also have a small IGD value, which may be own to

their advantages on diversity. In NSGA-III and MOEA/D, the diversity of the population is always kept well with uniform weight vectors. In terms of the DTLZ6, the results show that none of the algorithms performs well within a limited number of evaluations. From the above analysis and the results in Table 4, we can draw the conclusion that the TSMTF has a significant advantage in DTLZ1 and DTLZ3 in MaOPs.

As we can see it at a glance, for all the algorithms, DTLZ2 and DTLZ4 are easier than DTLZ1 and DTLZ3. In DTLZ2 and DTLZ4 problems with 100 decision variables, S3CMAES can converge and its performance is better than half of the compared algorithms, while it cannot converge on DTLZ1 and DTLZ3. With the number of decision variables increases, as the experimental results shown, S3CMAES degenerates its performance due to the limitation of function evaluations, because its grouping strategy and searching in sub-populations require a large number of function evaluations. As for CMOPSO, it shows little advantages in this experiments on MaOPs due to its particles' learning strategy. Similar to the other compared algorithms, it performs the best on DTLZ2 and DTLZ4. Since DTLZ1 and DTLZ3 is so complex that CMOPSO is unable to converge. For LMEA, as shown in Tables 4, 5 and 6, it cannot perform so well, while the other algorithms reached their convergency state. It seems to need extra function evaluations and it behaves worse as the number of decision variables increases. NSGA-II has the same issue. Within a limited number of evaluations, compared with the other algorithms, NSGA-II cannot converge for most MaOPs, but it shows better performance occasion-

Table 4 IGD results of MOEA/D, NSGA-II, NSAG-III, KnEA, LMEA, and TSMITF on DTLZ problems with 100 decision variables

Problem	M	D	MOEA/D	KnEA	LMEA	NSGA-II	NSGA-III	S3CMAES	CMOPSO	TSMITF
DTLZ1	3	100	3.0442e+2 (3.37e+1) – 2.4087e+2 (2.75e+1) – 2.7651e+3 (9.93e+1) – 4.0556e+2 (5.57e+1) – 5.4262e+2 (6.99e+1) – 8.6729e+2 (8.43e+1) – 1.2540e+3 (2.16e+2) – 1.9095e–1 (1.66e–1)							
DTLZ1	5	100	2.6857e+2 (4.65e+1) – 6.3513e+2 (6.27e+1) – 2.2926e+3 (1.16e+2) – 1.7649e+3 (1.08e+2) – 8.3334e+2 (4.89e+1) – 5.7988e+2 (2.74e+1) – 1.9695e+3 (1.74e+2) – 2.6679e–1 (2.19e–1)							
DTLZ1	10	100	1.3675e+2 (2.01e+1) – 9.0416e+2 (6.60e+1) – 1.9725e+3 (1.58e+2) – 2.3400e+3 (1.28e+2) – 6.1317e+2 (9.48e+1) – 3.9624e+2 (2.15e+1) – 1.8423e+3 (6.67e+1) – 2.6186e–1 (9.23e–2)							
DTLZ2	3	100	1.1632e–1 (1.31e–2) – 6.8051e–2 (1.92e–3) + 6.6112e+0 (2.82e–1) – 9.4672e–2 (5.83e–3) + 7.5303e–2 (5.19e–3) + 2.0941e+0 (3.83e–1) – 1.0278e–1 (7.18e–3) ≈ 1.1166e–1 (2.11e–2)							
DTLZ2	5	100	3.5825e–1 (5.11e–2) – 2.4237e–1 (9.48e–3) + 6.5715e+0 (2.38e–1) – 1.9783e+0 (3.96e–1) – 3.6429e–1 (3.00e–2) – 3.2644e–1 (9.45e–2) – 2.9763e+0 (5.37e–1) – 3.2963e–1 (3.04e–2)							
DTLZ2	10	100	4.5929e–1 (1.32e–2) + 7.6984e–1 (2.44e–1) + 6.3815e+0 (2.19e–1) – 8.8040e+0 (2.01e+0) – 7.1513e–1 (7.18e–2) + 7.1696e–1 (1.21e–1) + 1.3523e+1 (3.05e+0) – 9.2663e–1 (9.32e–2)							
DTLZ3	3	100	9.8922e+2 (9.47e+1) – 6.7821e+2 (7.08e+1) – 8.9683e+3 (2.34e+2) – 7.2844e+2 (9.90e+1) – 1.3513e+3 (1.95e+2) – 3.0516e+3 (2.37e+2) – 3.4503e+3 (6.98e+2) – 6.7821e–1 (4.68e–1)							
DTLZ3	5	100	1.1909e+3 (1.78e+2) – 1.5893e+3 (1.81e+2) – 8.5829e+3 (3.11e+2) – 5.3547e+3 (3.26e+2) – 2.6588e+3 (1.91e+2) – 2.4131e+3 (5.48e+1) – 5.5396e+3 (7.04e+2) – 8.0925e–1 (4.14e–1)							
DTLZ3	10	100	6.4614e+2 (1.01e+2) – 4.1133e+3 (2.50e+2) – 8.0781e+3 (2.31e+2) – 1.0394e+4 (1.28e+3) – 1.9349e+3 (4.05e+2) – 2.2826e+3 (4.94e+1) – 7.5365e+3 (4.10e+2) – 1.0043e+0 (2.66e–1)							
DTLZ4	3	100	5.7819e–1 (3.27e–1) + 1.3224e–1 (1.65e–1) + 6.8992e+0 (3.13e–1) – 1.6011e–1 (1.54e–1) + 1.1459e–1 (1.18e–1) + 2.3543e+0 (6.27e–1) – 2.0841e–1 (2.03e–1) + 9.6252e–1 (2.94e–3)							
DTLZ4	5	100	7.6142e–1 (1.96e–1) + 2.9927e–1 (7.16e–2) + 6.7935e+0 (2.43e–1) – 1.6433e+0 (2.99e–1) – 4.7655e–1 (5.26e–2) + 7.4879e–1 (1.12e–1) + 2.7014e+0 (5.99e–1) – 1.1332e+0 (5.09e–3)							
DTLZ4	10	100	8.6493e–1 (1.72e–1) + 5.1881e–1 (2.59e–2) + 6.4650e+0 (3.04e–1) – 7.6857e+0 (6.87e–1) – 7.0442e–1 (1.42e–1) + 9.0166e–1 (6.29e–2) + 8.0543e+0 (7.95e–1) – 1.0204e+0 (8.40e–2)							
DTLZ5	3	100	4.0881e–2 (6.31e–3) ≈ 2.3787e–2 (7.99e–3) + 6.4868e+0 (2.65e–1) – 3.0485e–2 (4.10e–3) + 4.1985e–2 (7.13e–3) ≈ 2.0666e+0 (4.01e–1) – 5.8371e–2 (7.78e–3) – 4.1641e–2 (8.07e–3)							
DTLZ5	5	100	3.7574e–2 (3.34e–3) + 5.0541e–1 (9.09e–2) – 6.3878e+0 (2.55e–1) – 5.2016e+0 (4.66e–1) – 6.0844e–1 (6.68e–2) – 2.7503e–1 (1.70e–1) – 6.1729e+0 (2.03e+0) – 5.1708e–2 (1.44e–2)							
DTLZ5	10	100	6.5735e–2 (2.03e–2) – 1.7157e+0 (2.19e–1) – 4.2787e+0 (2.64e+0) – 7.0954e+0 (8.19e–1) – 1.4381e+0 (2.46e–1) – 3.1411e–1 (2.38e–1) – 1.4234e+1 (3.49e+0) – 2.3729e–2 (9.69e–3)							
DTLZ6	3	100	4.2192e+1 (2.59e+0) – 2.5128e+1 (3.68e–1) – 8.7038e+1 (3.68e–1) – 3.3337e+1 (2.31e+0) – 2.9662e+1 (2.00e+0) – 7.0404e+1 (1.95e+0) – 4.6338e+1 (4.19e+0) – 1.3122e+1 (2.58e+0)							
DTLZ6	5	100	5.1681e+1 (2.34e+0) – 4.1939e+1 (2.67e+0) – 8.5151e+1 (3.97e–1) – 7.7506e+1 (1.47e+0) – 6.1686e+1 (1.68e+0) – 6.9925e+1 (1.72e+0) – 8.4553e+1 (1.61e+0) – 2.8037e+1 (3.26e+0)							
DTLZ6	10	100	4.8786e+1 (3.01e+0) – 4.3390e+1 (3.28e+0) ≈ 8.0610e+1 (3.68e–1) – 7.7589e+1 (8.85e–1) – 6.2870e+1 (1.96e+0) – 6.5101e+1 (2.16e+0) – 8.0728e+1 (1.17e+0) – 4.1285e+1 (4.85e+0)							
+/-/≈			5/12/1	7/10/1	0/18/0	3/15/0	5/12/1	3/15/0	1/16/1	

Table 5 IGD results of MOEA/D, NSGA-II, NSGA-III, KnEA, LMEA, and TSMITF on DTLZ problems with 200 decision variables

Problem	M	D	MOEA/D	KnEA	LMEA	NSGA-II	NSGA-III	S3CMAES	CMOPSO	TSMITF
DTLZ1	3	200	9.3145e+2 (9.02e+1) – 2.5301e+3 (1.20e+2) – 2.5488e+3 (8.80e+1) – 1.7564e+3 (1.58e+2) – 5.7167e+3 (1.66e+2) – 5.5067e+3 (1.60e+2) – 2.3778e+3 (2.25e+2) – 3.9263e–1 (5.25e–1)							
DTLZ1	5	200	7.3058e+2 (1.04e+2) – 4.1275e+3 (1.97e+2) – 1.9729e+3 (1.22e+2) – 1.7772e+3 (1.48e+2) – 4.7541e+3 (1.75e+2) – 2.4845e+3 (2.21e+2) – 3.7921e+3 (4.59e+2) – 3.9704e–1 (4.02e–1)							
DTLZ1	10	200	3.0087e+3 (1.04e+2) – 4.6185e+3 (3.11e+2) – 3.6294e+2 (3.80e+1) – 2.0562e+3 (1.44e+2) – 4.1168e+3 (2.01e+2) – 1.1594e+3 (7.60e+1) – 3.7292e+3 (1.02e+2) – 3.9067e–1 (4.12e–1)							
DTLZ2	3	200	8.3186e–1 (1.89e–1) – 2.1955e–1 (1.97e–2) – 2.8158e–1 (1.88e–2) – 1.4158e–1 (1.07e–2) + 1.3839e+1 (3.04e–1) – 1.5630e+1 (6.25e–1) – 1.9966e–1 (1.71e–2) – 1.5868e–1 (2.71e–2)							
DTLZ2	5	200	8.5248e–1 (2.28e–1) – 5.9032e+0 (7.39e–1) – 6.7520e–1 (5.04e–2) – 3.8741e–1 (3.54e–2) – 1.4029e+1 (4.75e–1) – 1.6539e+1 (1.08e+0) – 4.4760e+0 (4.91e–1) – 3.2017e–1 (2.38e–2)							
DTLZ2	10	200	3.2685e+0 (2.85e–1) – 1.5765e+1 (1.89e+0) – 1.9930e+0 (9.11e–1) – 1.4709e+0 (2.79e–1) – 1.3685e+1 (4.66e–1) – 3.6572e+0 (4.52e–1) – 2.3657e+1 (5.85e+0) – 6.9526e–1 (3.32e–2)							
DTLZ3	3	200	3.0645e+3 (2.10e+2) – 5.0909e+3 (3.48e+2) – 7.4995e+3 (3.36e+2) – 4.2359e+3 (4.17e+2) – 1.8439e+4 (4.55e+2) – 1.9131e+4 (6.84e+2) – 7.2260e+3 (9.31e+2) – 1.0958e+0 (1.06e+0)							
DTLZ3	5	200	3.6352e+3 (4.47e+2) – 1.4491e+4 (8.83e+2) – 7.0179e+3 (4.30e+2) – 5.1406e+3 (4.63e+2) – 1.8513e+4 (4.04e+2) – 1.0373e+4 (6.00e+2) – 1.1492e+4 (1.47e+3) – 9.2110e–1 (1.28e+0)							
DTLZ3	10	200	1.1807e+4 (4.36e+2) – 2.1599e+4 (1.23e+3) – 2.4820e+3 (9.19e+2) – 9.7553e+3 (6.67e+2) – 1.7913e+4 (3.77e+2) – 5.9652e+3 (3.59e+2) – 1.7016e+4 (4.87e+2) – 9.9869e–1 (2.49e–1)							
DTLZ4	3	200	1.8602e+0 (1.16e+0) – 2.5688e–1 (1.90e–2) + 3.2747e–1 (2.24e–2) + 1.6580e–1 (1.78e–2) – 1.4061e+1 (3.19e–1) – 1.4759e+1 (3.42e+0) – 4.2364e–1 (1.21e–1) + 9.8956e–1 (1.09e–2)							
DTLZ4	5	200	7.1593e–1 (1.83e–1) + 4.6721e+0 (7.44e–1) – 9.0910e–1 (9.36e–2) – 5.5208e–1 (5.47e–2) + 1.4184e+1 (3.99e–1) – 5.0059e+0 (1.90e+0) – 4.9235e+0 (5.88e–1) – 1.1444e+0 (8.89e–3)							
DTLZ4	10	200	3.650e+0 (4.75e–1) – 1.5316e+1 (7.55e–1) – 8.6144e–1 (1.29e–1) + 1.1129e+0 (1.48e–1) – 1.3762e+1 (4.38e–1) – 1.1049e+0 (1.23e–1) – 1.6094e+1 (1.52e+0) – 8.9596e–1 (8.36e–2)							
DTLZ5	3	200	2.8403e–1 (9.97e–2) – 1.6537e–1 (1.99e–2) – 2.7552e–1 (2.07e–2) – 1.2970e–1 (1.39e–2) – 1.3689e+1 (3.81e–1) – 1.5810e+1 (5.92e–1) – 2.3900e–1 (3.67e–2) – 1.1647e–1 (1.83e–2)							
DTLZ5	5	200	9.3847e–2 (3.70e–2) – 1.1756e+1 (6.27e–1) – 1.0673e+0 (9.46e–2) – 6.0708e–1 (8.98e–2) – 1.3781e+1 (4.31e–1) – 1.6238e+1 (7.92e–1) – 9.2033e+0 (9.38e–1) – 7.2253e–2 (1.27e–2)							
DTLZ5	10	200	3.5825e+0 (2.99e–1) – 1.4984e+1 (9.88e–1) – 5.9384e–1 (7.53e–1) – 3.4473e+0 (3.82e–1) – 1.3358e+1 (3.36e–1) – 3.6998e+0 (5.33e–1) – 2.6803e+1 (7.01e+0) – 3.3920e–2 (5.52e–3)							
DTLZ6	3	200	1.2634e+2 (2.50e+0) – 1.0746e+2 (1.93e+0) – 1.0682e+2 (2.29e+0) – 8.4978e+1 (1.87e+0) – 1.7683e+2 (5.79e–1) – 1.7134e+2 (1.28e+0) – 9.7376e+1 (6.41e+0) – 5.2788e+1 (6.14e+0)							
DTLZ6	5	200	1.1644e+2 (3.68e+0) – 1.6338e+2 (1.94e+0) – 1.3153e+2 (2.31e+0) – 1.0817e+2 (2.37e+0) – 1.7510e+2 (4.52e–1) – 1.4458e+2 (3.49e+0) – 1.7138e+2 (2.50e+0) – 6.4504e+1 (5.79e+0)							
DTLZ6	10	200	1.5417e+2 (1.67e+0) – 1.6570e+2 (1.12e+0) – 1.3838e+2 (2.20e+0) – 1.2099e+2 (3.36e+0) – 1.7046e+2 (4.60e–1) – 1.4213e+2 (3.28e+0) – 1.6892e+2 (1.44e+0) – 8.0260e+1 (8.06e+0)							
+/-/∞			1/17/0	1/17/0	2/16/0	3/15/0	0/18/0	0/18/0	1/17/0	

Table 6 IGD results of MOEA/D, NSGA-II, NSGA-III, KnEA, LMEA, and TSMITF on DTLZ problems with 500 decision variables

Problem	M	D	MOEA/D	KnEA	LMEA	NSGA-II	NSGA-III	S3CMAES	CMOPSO	TSMITF
DTLZ1	3	500	9.4482e+3 (1.59e+2) – 4.4814e+3 (4.44e+2) – 9.9297e+3 (2.40e+2) – 7.8466e+3 (2.96e+2) – 8.14802e+4 (2.41e+2) – 1.9721e+4 (3.24e+3) – 5.3649e+3 (4.16e+2) – 2.9675e–1 (3.33e–1)							
DTLZ1	5	500	5.0617e+3 (7.87e+2) – 1.1474e+4 (3.64e+2) – 8.1828e+3 (1.74e+2) – 6.3261e+3 (2.08e+2) – 1.2050e+4 (2.35e+2) – 1.2233e+4 (2.15e+2) – 6.1949e+3 (5.01e+2) – 2.1903e–1 (1.91e–1)							
DTLZ1	10	500	1.5817e+3 (1.77e+2) – 1.1442e+4 (5.72e+2) – 7.7090e+3 (1.06e+2) – 5.2028e+3 (2.25e+2) – 1.0400e+4 (5.59e+2) – 8.9332e+3 (2.36e+2) – 9.4095e+3 (3.07e+2) – 2.8760e–1 (3.03e–1)							
DTLZ2	3	500	4.8181e+0 (1.53e+0) – 7.3534e–1 (4.77e–2) – 1.3223e+0 (6.61e–2) – 6.0212e–1 (4.70e–2) – 3.6849e+1 (4.46e–1) – 4.1845e+1 (1.35e+0) – 1.0514e+0 (1.04e–1) – 2.0839e–1 (3.49e–2)							
DTLZ2	5	500	4.2977e+0 (1.03e+0) – 1.3462e+1 (9.72e–1) – 2.1493e+0 (1.08e–1) – 5.4657e–1 (3.67e–2) – 3.6584e+1 (5.78e–1) – 3.9243e+1 (1.23e+0) – 8.9134e+0 (8.01e–1) – 2.1777e–1 (1.80e–2)							
DTLZ2	10	500	6.8704e+0 (2.54e+0) – 4.3650e+1 (7.20e+0) – 8.7791e+0 (4.07e–1) – 9.9453e–1 (1.00e–1) + 3.6701e+1 (6.65e–1) – 3.9573e+1 (1.04e+0) – 5.7139e+1 (1.52e+1) – 9.9848e–1 (1.28e–1)							
DTLZ3	3	500	2.3877e+4 (4.03e+2) – 1.5724e+4 (1.17e+3) – 2.6394e+4 (9.78e+2) – 2.4011e+4 (9.84e+2) – 4.9231e+4 (5.71e+2) – 5.3751e+4 (1.51e+3) – 1.8742e+4 (1.43e+3) – 1.9276e+0 (2.46e+0)							
DTLZ3	5	500	2.3877e+4 (3.24e+3) – 4.1805e+4 (7.75e+2) – 3.2502e+4 (5.53e+2) – 2.3908e+4 (1.79e+3) – 4.8928e+4 (6.16e+2) – 5.1694e+4 (1.31e+3) – 1.7400e+4 (2.87e+3) – 1.0959e+0 (2.48e+0)							
DTLZ3	10	500	1.7875e+4 (4.07e+3) – 5.6642e+4 (2.87e+3) – 3.6034e+4 (1.10e+3) – 2.5156e+4 (1.15e+3) – 4.8612e+4 (6.27e+2) – 5.1114e+4 (9.81e+2) – 4.5702e+4 (5.11e+2) – 1.0554e+0 (6.02e–1)							
DTLZ4	3	500	1.5115e+0 (7.19e–2) – 1.3771e+1 (4.89e+0) – 8.3924e–1 (5.06e–2) – 7.1653e–1 (4.25e–2) + 3.7145e+1 (6.42e–1) – 4.2404e+1 (1.43e+0) – 2.4848e+0 (3.57e–1) – 1.0291e+0 (2.42e–2)							
DTLZ4	5	500	4.4358e+0 (6.20e+0) – 1.1129e+1 (1.15e+0) – 2.8138e+0 (1.50e–1) – 7.4390e–1 (4.65e–2) + 3.7002e+1 (6.80e–1) – 1.1937e+1 (4.10e+0) – 1.2276e+1 (1.20e+0) – 1.1496e+0 (1.16e–2)							
DTLZ4	10	500	8.3595e–1 (1.20e–1) + 4.0348e+1 (1.99e+0) – 1.4473e+1 (5.44e–1) – 3.3645e+0 (3.59e–1) – 3.6681e+1 (8.20e–1) – 7.6893e+0 (3.42e+0) – 4.0385e+1 (4.82e+0) – 1.4155e+0 (1.35e–1)							
DTLZ5	3	500	1.5851e+0 (8.50e–2) – 3.4762e+0 (9.73e–1) – 8.4330e–1 (5.46e–2) – 6.9939e–1 (4.64e–2) – 3.6648e+1 (5.74e–1) – 4.2024e+1 (1.78e+0) – 2.6326e+0 (4.00e–1) – 1.7894e–1 (1.79e–2)							
DTLZ5	5	500	2.7511e+0 (1.59e+0) – 3.2352e+1 (1.11e+0) – 3.5646e+0 (1.56e–1) – 9.2132e–1 (4.67e–1) – 3.6612e+1 (5.67e–1) – 3.9264e+1 (1.52e+0) – 2.1893e+1 (1.27e+0) – 7.5147e–2 (9.83e–3)							
DTLZ5	10	500	1.0002e+1 (3.51e+0) – 3.9100e+1 (1.34e+0) – 1.5026e+1 (1.29e+0) – 3.6777e+0 (4.43e–1) – 3.6214e+1 (5.04e–1) – 3.9067e+1 (1.20e+0) – 7.4239e+1 (1.77e+1) – 6.5108e–2 (1.24e–2)							
DTLZ6	3	500	3.4083e+2 (2.63e+0) – 3.7017e+2 (3.22e+0) – 3.4702e+2 (2.39e+0) – 2.8931e+2 (2.70e+0) – 4.4713e+2 (5.49e–1) – 4.5344e+2 (1.81e+0) – 2.5107e+2 (1.04e+1) – 1.3708e+2 (2.04e+1)							
DTLZ6	5	500	3.6635e+2 (3.23e+0) – 4.2441e+2 (2.37e+0) – 3.8316e+2 (2.18e+0) – 3.0361e+2 (2.19e+0) – 4.4537e+2 (5.90e–1) – 4.4633e+2 (1.55e+0) – 4.3685e+2 (3.13e+0) – 1.5003e+2 (1.53e+1)							
DTLZ6	10	500	3.9486e+2 (4.66e+0) – 4.3093e+2 (2.34e+0) – 4.1303e+2 (3.70e+0) – 3.5426e+2 (3.44e+0) – 4.4100e+2 (6.85e–1) – 4.4376e+2 (1.42e+0) – 4.3299e+2 (2.73e+0) – 1.9363e+2 (3.09e+1)							
+/-/∞			1/17/0	0/18/0	0/18/0	3/15/0	0/18/0	0/18/0	0/18/0	0/18/0

Table 7 IGD Results of KnEA, LMEA, and TSMTF on LSMOP problems with 3 objectives

Problem	<i>M</i>	<i>D</i>	KnEA	LMEA	TSMTF
LSMOP1	3	100	2.2781e−1 (5.31e−2) +	3.9175e−1 (7.04e−1) +	8.3099e−1 (4.71e−3)
LSMOP1	3	200	7.3984e−1 (2.01e−1) +	1.0085e+1 (4.38e−1) −	8.3911e−1 (2.30e−3)
LSMOP1	3	500	1.8582e+0 (5.26e−1) −	1.0773e+1 (5.08e−1) −	8.4140e−1 (1.89e−3)
LSMOP2	3	100	1.2367e−1 (2.05e−3) +	1.1034e−1 (4.75e−2) +	1.3800e−1 (4.16e−3)
LSMOP2	3	200	9.1770e−2 (1.62e−3) −	1.2463e−1 (9.83e−4) −	9.0664e−2 (1.29e−2)
LSMOP2	3	500	5.3127e−2 (5.11e−4) +	6.7379e−2 (1.48e−3) +	8.7213e−2 (2.27e−2)
LSMOP3	3	100	7.7722e−1 (1.65e−1) +	2.7517e+0 (1.77e+0) −	8.5910e−1 (4.71e−3)
LSMOP3	3	200	2.8160e+0 (8.52e−1) −	2.8474e+2 (2.84e+2) −	8.6010e−1 (7.76e−4)
LSMOP3	3	500	8.3796e+0 (1.10e+0) −	2.3411e+2 (1.82e+2) −	8.6031e−1 (5.46e−4)
LSMOP4	3	100	2.7838e−1 (1.11e−2) +	2.8113e−1 (8.74e−2) +	8.6072e−1 (5.90e−6)
LSMOP4	3	200	2.3674e−1 (4.35e−3) +	3.8725e−1 (4.82e−3) −	3.4893e−1 (1.72e−2)
LSMOP4	3	500	1.3116e−1 (1.62e−3) +	2.0595e−1 (1.80e−3) −	1.6945e−1 (7.09e−3)
LSMOP5	3	100	5.1282e−1 (1.33e−1) +	3.2654e+0 (3.92e+0) −	5.4246e−1 (8.70e−3)
LSMOP5	3	200	3.2552e+0 (3.12e−1) −	1.7028e+1 (1.37e+0) −	6.4134e−1 (4.54e−2)
LSMOP5	3	500	6.9863e+0 (8.28e−1) −	1.9003e+1 (9.92e−1) −	7.1885e−1 (5.90e−2)
LSMOP6	3	100	1.4197e+0 (6.76e−1) −	1.3746e+3 (3.48e+3) −	9.3643e−1 (1.81e−1)
LSMOP6	3	200	1.8257e+1 (1.72e+1) −	1.8257e+1 (1.72e+1) −	1.0700e+0 (2.67e−1)
LSMOP6	3	500	3.0170e+2 (3.10e+2) −	3.3544e+4 (4.81e+3) −	1.0173e+0 (3.07e−1)
+/-/≈			9/9/0	3/15/0	

Table 8 IGD results of KnEA, LMEA, and TSMTF on LSMOP problems with 5 objectives

Problem	<i>M</i>	<i>D</i>	KnEA	LMEA	TSMTF
LSMOP1	5	100	4.4122e−1 (1.44e−1) +	5.2090e−1 (8.26e−1) +	8.7298e−1 (3.69e−2)
LSMOP1	5	200	8.0131e−1 (3.65e−1) ≈	8.5327e+0 (1.29e+0) −	9.1906e−1 (1.30e−2)
LSMOP1	5	500	2.3319e+0 (7.03e−1) −	1.0186e+1 (5.47e−1) −	9.3770e−1 (2.68e−3)
LSMOP2	5	100	2.3723e−1 (3.66e−2) −	2.3167e−1 (8.13e−2) +	3.5016e−1 (3.92e−2)
LSMOP2	5	200	1.8554e−1 (5.02e−3) +	2.6465e−1 (1.89e−2) −	2.5170e−1 (1.57e−2)
LSMOP2	5	500	1.3768e−1 (2.69e−3) +	1.8904e−1 (8.73e−3) −	1.8141e−1 (7.09e−3)
LSMOP3	5	100	1.2072e+0 (4.61e−1) −	6.4242e+0 (1.01e+1) −	9.0503e−1 (1.64e−2)
LSMOP3	5	200	6.0803e+0 (2.32e+0) −	8.0316e+2 (8.72e+2) −	9.2634e−1 (2.32e−2)
LSMOP3	5	500	1.2123e+1 (3.02e+0) −	1.5857e+3 (9.02e+2) −	9.5386e−1 (9.82e−3)
LSMOP4	5	100	4.4841e−1 (1.28e−1) +	4.6621e−1 (9.21e−2) +	8.8824e−1 (1.44e−1)
LSMOP4	5	200	4.2685e−1 (7.45e−2) +	5.9842e−1 (7.16e−2) −	5.1472e−1 (3.62e−2)
LSMOP4	5	500	4.3046e−1 (1.13e−1) −	3.6109e−1 (1.71e−2) −	3.3690e−1 (1.73e−2)
LSMOP5	5	100	1.0390e+0 (1.30e−1) −	8.5515e−1 (7.17e−1) −	4.8722e−1 (3.04e−2)
LSMOP5	5	200	2.5155e+0 (1.18e+0) −	9.2917e+0 (2.98e+0) −	5.0204e−1 (2.70e−2)
LSMOP5	5	500	6.0354e+0 (2.25e+0) −	1.4148e+1 (1.37e+0) −	5.1161e−1 (3.43e−2)
LSMOP6	5	100	2.3221e+0 (7.87e−1) −	9.9807e−1 (3.16e−1) ≈	1.0382e+0 (5.55e−2)
LSMOP6	5	200	1.6889e+0 (5.97e−1) +	4.8018e+3 (3.05e+3) −	1.0195e+1 (9.80e+0)
LSMOP6	5	500	7.0396e+1 (4.27e+1) −	1.6902e+4 (4.11e+3) −	1.4058e+1 (1.55e+1)
+/-/≈			6/11/1	3/14/1	

Table 9 IGD results of KnEA, LMEA, and TSMTF on LSMOP problems with 10 objectives

Problem	<i>M</i>	<i>D</i>	KnEA	LMEA	TSMTF
LSMOP1	10	300	1.8435e+0 (5.25e−1) −	9.0316e+0 (1.13e+0) −	9.9479e−1 (5.64e−3)
LSMOP1	10	500	3.8685e+0 (8.62e−1) −	9.7671e+0 (7.75e−1) −	1.0047e+0 (2.97e−3)
LSMOP2	10	300	3.7888e−1 (9.57e−3) +	5.5187e−1 (9.98e−2) −	4.1819e−1 (2.84e−2)
LSMOP2	10	500	3.9559e−1 (7.38e−2) −	5.0921e−1 (3.04e−2) −	3.4442e−1 (2.26e−2)
LSMOP3	10	300	2.3763e+2 (3.75e+2) −	3.5795e+3 (1.93e+3) −	9.3597e+0 (2.61e+1)
LSMOP3	10	500	4.0084e+2 (4.64e+2) −	5.0309e+3 (2.56e+3) −	1.6290e+0 (3.39e−1)
LSMOP4	10	300	4.5887e−1 (1.55e−2) +	6.2192e−1 (4.99e−2) −	5.3409e−1 (3.83e−2)
LSMOP4	10	500	3.8021e−1 (1.57e−2) +	5.2533e−1 (1.84e−2) −	4.5271e−1 (3.39e−2)
LSMOP5	10	300	1.9322e+1 (4.25e+0) −	1.4527e+1 (3.10e+0) −	1.1452e+0 (8.75e−2)
LSMOP5	10	500	1.3372e+1 (3.34e+0) −	1.4857e+1 (9.23e−1) −	1.0533e+0 (9.63e−2)
LSMOP6	10	300	2.2281e+4 (2.65e+4) −	1.1117e+3 (1.01e+3) −	1.5714e+0 (3.63e−1)
LSMOP6	10	500	3.0241e+4 (2.17e+4) −	8.1474e+2 (1.21e+3) −	1.3910e+0 (1.92e−1)
+ / − / ≈			3/9/0	0/12/0	

ally, while the TSMTF can converge. As for NSGA-III, in DTLZ2, compared with TSMTF, it performs better occasionally, in DTLZ4 with 100D and 200D decision variables, it nearly always performs better than TSMTF, except DTLZ4. Their difference is getting large with the number of decision variables increases. The performance of MOEA/D is similar to NSGA-III, both of them performs better than TSMTF occasionally on DTLZ2 with 200 decision variables. TSMTF shows advantages in DTLZ4 with 100 decision variables, with the number of decision variables increase. As for KnEA, its IGD results are better than TSMTF in terms of several experimental results on DTLZ2 and DTLZ4. TSMTF shows the best performance on DTLZ1 and DTLZ3 in each experiment. On DTLZ6, all algorithms show similar performance.

From the above results on DTLZ problems in Tables 4, 5 and 6, KnEA, LMEA, and TSMTF are good at solving LSMaOPs. Therefore, we further compare them on large-scale benchmark problems LSMOP [34] with different numbers of decision variables and objective functions. The IGD results are shown in Tables 7, 8 and 9

In Tables 7, 8 and 9, in general, in most cases, the proposed framework performs better than the other two algorithms, especially on 10-objective problems in Table 9. TSMTF shows disadvantage in some situations. Three-objective LSMOP4 is a easy-converging problem, and all those algorithms converge to the true PF and TSMTF cannot outperform the other two algorithms. However, for the other 3-objective LSMOP problems, neither KnEA and LMEA cannot converge, while TSMTF has better convergence ability, which make TSMTF a best-performing algorithm.

KnEA performs the best on 3-objective LSMOP problems. The results on LSMOP1, LSMOP2, and LSMOP4 show that KnEA converges in almost all cases. As the number of objectives increases, Tables 7, 8 and 9 show that the advantage of

KnEA decreases, but it still performs the best on 10-objective LSMOP4 compared with the other two algorithms. As for LMEA, the results shows that it can converge on LSMOP2 and LSMOP4. LSMOP6 is too hard for LMEA. LMEA is able to obtain the solutions near the true PF in the other four LSMOPs. As the number of decision variables increases, the performance of LMEA is getting worse. The performance of TSMTF tells that it converges to the true PF in almost all cases. When TSMTF is not better than the other two algorithms, the results shows that their performance difference is tiny. While TSMTF performs the best, the results shows that the other two algorithms are far from the true PF.

The results in Tables 1, 2, 3, 4, 5, 6, 7, 8 and 9 show that TSMTF performs the best among all the algorithms. The main reason of the good performance is from the contribution of stage 1. In this stage, we transformed the high-dimensional search space into a low-dimensional space by a number of weight vectors, and then, the population in TSMTF evolves by optimizing those weight vectors. In this way, with the decreasing number of decision variables, the searching cost (number of function evaluations) has been greatly reduced. To further improve the diversity, TSMTF specially increase the diversity of a small number of the individuals with good convergence. Stage 2 can increase the diversity and convergence simultaneously. After stage 2, we employ NSGA-III to make the population more evenly distributed. LMEA costs too many evaluations on decision variable grouping and S3CMAES costs much on the searching in sub-populations, so their performance degenerates when the computation budget is limited. Since KnEA, NSGA-III, CMOPSO, and MOEA/D are easily get trapped in a high-dimensional decision space without any dimension reduction, it cannot find the true PS for most large-scale MaOPs.

Conclusions and remarks

In this work, we propose a two-stage framework combined with MFEA to address large-scale MaOPs. In stage 1, we make use of the bi-directional search strategy combined with MFEA. In other words, we transformed the MaOP into a multi-tasking optimization problem. After that, it turns to stage 2, which aims to improve the diversity of the population. We apply multi-tasking to a number of sub-problems from the multi-objective optimization problem. At last, the population is optimized as NSGA-III. As the experiments shows, with limited number of function evaluations, the proposed algorithm outperforms compared algorithms on MaOPs, especially on complex problems such as DTLZ1 and DTLZ3. The results verify the effectiveness of this framework.

In general, the TSMTF shows well performance and can still be improved in both stages, and there is still much room for improvement. For example, the transformed objective function in stage 1, we adopt the bi-directional strategy, and we need to think about other effective methods. In addition, the diversity in stage 2 still needs to be improved.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Tian Y, Wang H, Zhang X, Jin Y (2017) Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization. *Complex Intell Syst* 3(4):247–263
2. Coello CAC, Brambila SG, Gamboa JF, Tapia MGC, Gómez RH (2020) Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex Intell Syst* 6(2):221–236
3. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
4. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. *TIK-report* 103
5. Zhang Q, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
6. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *International conference on parallel problem solving from nature*. Springer, Berlin, Heidelberg, pp 832–842
7. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multi-objective selection based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669
8. Bader J, Zitzler E (2011) HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput* 19(1):45–76
9. 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
10. Wang R, Purshouse RC, Fleming PJ (2012) Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Trans Evol Comput* 17:474–494
11. Zhang X, Tian Y, Jin Y (2014) A knee point driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(6):761–776
12. Cheng J, Yen GG, Zhang G (2015) A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Trans Evol Comput* 19(4):592–605
13. Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2017) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf Sci* 427:63–76
14. Tian Y, Zheng X, Zhang X, Jin Y (2019) Efficient large-scale multi-objective optimization based on a competitive swarm optimizer. *IEEE Trans Cybern* 50(8):3696–3708
15. Cheng R, Jin Y (2014) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
16. Gong M, Li H, Luo E, Liu J, Liu J (2016) A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing. *IEEE Trans Evol Comput* 21(2):234–248
17. Wang H, Jiao L, Shang R, He S, Liu F (2015) A memetic optimization strategy based on dimension reduction in decision space. *Evol Comput* 23(1):69–100
18. Wang H, Jin Y (2017) Efficient nonlinear correlation detection for decomposed search in evolutionary multi-objective optimization. In: *2017 IEEE Congress on evolutionary computation (CEC)*. IEEE, New York, pp 649–656
19. Ma X, Liu F, Qi Y, Wang X, Li L, Jiao L, Yin M, Gong M (2016) A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Trans Evol Comput* 20(2):275–298
20. Cheng R, Zhang X, Tian Y (2016) A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans Evol Comput* 22(1):99–112
21. Chen H, Cheng R, Wen J, Li H, Weng J (2020) Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. *Inf Sci* 509:457–469
22. He C, Li L, Tian Y, Zhang X, Cheng R, Jin Yaochu, Yao Xin (2019) Accelerating large-scale multiobjective optimization via problem reformulation. *IEEE Trans Evol Comput* 23(6):949–961
23. Zille H, Ishibuchi H, Mostaghim S, Nojima Y (2018) A framework for large-scale multiobjective optimization based on problem transformation. *IEEE Trans Evol Comput* 22(2):260–275
24. Gupta A, Ong YS, Feng L (2016) Multifactorial evolution: toward evolutionary multitasking. *IEEE Trans Evol Comput* 20(3):343–357
25. Tian Y, Wang H, Zhang X, Jin Y (2017) Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization. *Complex Intell Syst* 3(4):247–263
26. Zille H, Ishibuchi H, Mostaghim S, Nojima Y (2017) A framework for large-scale multiobjective optimization based on problem transformation. *IEEE Trans Evol Comput* 22(2):260–275
27. Deb K, Jain H (2013) 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

28. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: *Evolutionary multiobjective optimization*. Springer, London, pp 105–145
29. Cheng R, Jin Y, Olhofer M, Sendhoff B (2017) Test problems for large-scale multiobjective and many-objective optimization. *IEEE Trans Cybern* 47(12):4108–4121
30. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
31. Deb K (2001) *Multi-objective optimization using evolutionary algorithms*, vol 16. Wiley, New York
32. Deb K, Agrawal RB et al (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9(2):115–148
33. Wilcoxon Frank, Katti SK, Wilcox Roberta A (1970) Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics* 1:171–259
34. Cheng Ran, Jin Yaochu, Olhofer Markus et al (2016) Test problems for large-scale multiobjective and many-objective optimization. *IEEE transactions on cybernetics* 47(12):4108–4121

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.