Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A many-objective evolutionary algorithm based on decomposition with dynamic resource allocation for irregular optimization*

Ming-gang DONG[1,2], Bao LIU[1], Chao JING[†‡1,2,3]

[1]College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China

[2]Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin 541004, China

[3]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic & Technology, Guilin 541004, China

[†]E-mail: jingchao@glut.edu.cn

Received June 28, 2019; Revision accepted Dec. 2, 2019; Crosschecked July 21, 2020

Abstract: The multi-objective optimization problem has been encountered in numerous fields such as high-speed train head shape design, overlapping community detection, power dispatch, and unmanned aerial vehicle formation. To address such issues, current approaches focus mainly on problems with regular Pareto front rather than solving the irregular Pareto front. Considering this situation, we propose a many-objective evolutionary algorithm based on decomposition with dynamic resource allocation (MaOEA/D-DRA) for irregular optimization. The proposed algorithm can dynamically allocate computing resources to different search areas according to different shapes of the problem's Pareto front. An evolutionary population and an external archive are used in the search process, and information extracted from the external archive is used to guide the evolutionary population to different search regions. The evolutionary population evolves with the Tchebycheff approach to decompose a problem into several subproblems, and all the subproblems are optimized in a collaborative manner. The external archive is updated with the method of shift-based density estimation. The proposed algorithm is compared with five state-of-the-art many-objective evolutionary algorithms using a variety of test problems with irregular Pareto front. Experimental results show that the proposed algorithm outperforms these five algorithms with respect to convergence speed and diversity of population members. By comparison with the weighted-sum approach and penalty-based boundary intersection approach, there is an improvement in performance after integration of the Tchebycheff approach into the proposed algorithm.

Key words: Many-objective optimization problems; Irregular Pareto front; External archive; Dynamic resource allocation; Shift-based density estimation; Tchebycheff approach

https://doi.org/10.1631/FITEE.1900321 **CLC number:** TP391

1 Introduction

In most real-life applications, such as high-speed train head shape design (Zhang L et al., 2017), overlapping community detection (Wen et al., 2017), power dispatch (Zeng and Sun, 2014), and unmanned aerial vehicle formation (Ruan and Duan, 2020), there are many optimization problems with more than one objective, known as multi-objective optimization problems (MOPs). Because these objectives often conflict with each other, no single solution satisfies all the objectives to achieve optimum values at the same

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61563012, 61802085, and 61203109), the Guangxi Natural Science Foundation of China (Nos. 2014GXNSFAA118371, 2015GXNSFBA139260, and 2020GXNSFAA159038), the Guangxi Key Laboratory of Embedded Technology and Intelligent System Foundation (No. 2018A-04), and the Guangxi Key Laboratory of Trusted Software Foundation (Nos. kx202011 and kx201926)

ORCID: Ming-gang DONG, https://orcid.org/0000-0001-7078-3942; Chao JING, https://orcid.org/0000-0002-4695-8746

© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

time. As a result, there exists a set of solutions, named Pareto optimal solutions, which presents trade-offs for the different objectives. An MOP can be stated as

$$\begin{aligned} &\min\ \boldsymbol{F}(\boldsymbol{x})=\left(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),...,f_m(\boldsymbol{x})\right)\\ &\text{s.t. } \boldsymbol{x}=(x_1,x_2,...,x_D)\in\Omega, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is a decision variable vector, Ω is the nonempty decision variable space, $\boldsymbol{F}$: $\Omega\rightarrow\mathbb{R}^m$ consists of m objective vectors, and $\mathbb{R}^m$ is the objective space (Coello, 2006). Specifically, if an MOP has more than three objectives, it is called a many-objective optimization problem (MaOP) (Chand and Wagner, 2015; Li BD et al., 2015).

Let a solution $\boldsymbol{x}^1$ dominate another solution $\boldsymbol{x}^2$ iff

$$\begin{cases} f_i(\boldsymbol{x}^1)\le f_i(\boldsymbol{x}^2),\ \forall i=1,2,...,m,\\ f_j(\boldsymbol{x}^1)<f_j(\boldsymbol{x}^2),\ \exists j\in\{1,2,...,m\}. \end{cases} \tag{2}$$

A solution $\boldsymbol{x}^*$ is said to be Pareto optimal if there exist no other feasible solutions that dominate it. The union of Pareto optimal solutions is called the Pareto set (PS), while the corresponding objective vector union is termed the Pareto front (PF) (Cai et al., 2015).

Multi-objective evolutionary algorithms (MOEAs) can obtain a set of solutions in a single run. Due to this population-based property, MOEAs are well suited for solving MOPs and have been predominant in dealing with MOPs over the past two decades, e.g., the improved strength Pareto evolutionary algorithm (SPEA2) (Zitzler et al., 2001), the improved Pareto envelope based selection algorithm (PSEA-II) (Corne et al., 2001), the nondominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002), the indicator-based evolutionary algorithm (IBEA) (Zitzler and Künzli, 2004), the developed version of generalized differential evolution (GDE3) (Kukkonen and Lampinen, 2005), and multiple objective particle swarm optimization (MOPSO) (Coello and Lechuga, 2002). However, along with the increasing number of objectives, there exists a large nondominated fraction of population, and evaluation of the diversity measure becomes computationally expensive. Due to the loss of selection pressure and the difficulties in maintaining population diversity, MOEAs have experienced substantial difficulties in solving MaOPs (Purshouse and Fleming, 2007). Consequently, the many-objective evolutionary algorithm (MaOEA) is becoming an active research topic in evolutionary optimization.

Deb and Jain (2014) followed the NSGA-II framework and proposed a reference-point-based MaOEA (NSGA-III), wherein a set of evenly distributed solutions is predefined to guarantee population diversity and enhance the convergence speed of the algorithm. To solve single-, multiple-, and many-objective problems, Seada and Deb (2016) proposed a unified evolutionary algorithm U-NSGA-III, in which a new niching-based selection procedure is used to automatically degenerate to an efficient equivalent algorithm for each class. To automatically balance the convergence speed and the diversity of population members, Seada et al. (2019) proposed a multiphase many-objective evolutionary optimization algorithm B-NSGA-III based on the general outline of U-NSGA-III. Zhang HP and Hui (2019) proposed a cooperative bat-searching algorithm (MOCBA), which is extended from a single-objective optimizer to a many-objective optimizer using the balanceable fitness estimation methods to balance diversity and convergence. An MOEA based on decomposition (MOEA/D) decomposes an MOP into a number of single-objective optimization subproblems and then solves them using a cooperative approach (Zhang QF and Li, 2007). Considering the benefit of decomposition, MOEA/D can be used to solve both MOPs and MaOPs. A reference vector-guided evolutionary algorithm (RVEA) for many-objective optimization partitions the objective space into a number of small subspaces with a set of reference vectors and elucidates user preferences, to target a preferred subset of the whole PF (Cheng et al., 2016). RVEA also uses an adaptation strategy of reference vectors in solving problems where the objective functions are not normalized well. Li K et al. (2015) proposed a many-objective optimization evolutionary algorithm based on dominance and decomposition (MOEA/DD), which exploits the merits offered by both the dominance and decomposition approaches to balance the convergence speed and the diversity during the evolutionary process. He ZN and Yen (2016) presented an MaOEA with objective space reduction and diversity improvement (MaOEA-R&D), which leverages a novel design of two stages. The whole

population first quickly approaches a small number of boundary points near the true PF, and then a diversity improvement strategy is used to make these solutions well spread and distributed. Moreover, many other algorithms, such as bi-goal dominance (Li MQ et al., 2015), *L*-optimality (Zou et al., 2008), fuzzy dominance (Wang GP and Jiang, 2007), reference point based dominance (RP-dominance) (Elarbi et al., 2018), grid-based EA (GrEA) (Yang et al., 2013), and knee point driven EA (KnEA) (Zhang XY et al., 2015) focus on modifying the dominance relationship to increase the selection pressure toward the PF.

Despite the fact that most existing MaOEAs can be used to solve MaOPs, they are considered mainly with the problems of regular PF. In contrast, to solve the problems with irregular PF, it is difficult to reach satisfaction with these algorithms. We provide an example. Fig. 1 shows the final populations obtained by NSGA-III for two widely used MaOPs, DTLZ4 (Deb et al., 2005) and IDTLZ1 (Jain and Deb, 2014) with eight objectives. DTLZ4 has a regular PF, while the PF of IDTLZ1 is irregular. NSGA-III achieves an even distribution for DTLZ4. In comparison, the distribution of NSGA-III for IDTLZ1 is poor. This is because the irregular shape of the PF is disconnected, inverted, and mostly degenerate, which increases the difficulty in approaching the true PF.

To improve the performance of MaOEAs in terms of solving MaOPs with irregular PF, an MOEA with an enhanced inverted generational distance indicator has been proposed for better versatility with different objective numbers and PF shapes (Tian et al., 2018). This proposed MOEA uses an adaptation method to adjust a set of reference points based on the indicator contributions of candidate solutions in an external archive (EA). Liu et al. (2017) proposed a reference points based MaOEA, in which a series of reference points with good performance is continuously generated according to the current population to guide the population evolution. An MOEA has been proposed to solve constrained MaOPs (Jain and Deb, 2014), which adopts an adaptive method that adds and deletes reference points depending on the crowding of candidates on different parts of the current nondominated front. Although these reference point adaptation methods have achieved significantly better performance for some MaOPs with irregular PF, they cannot effectively approach the true PFs with both high convergence speed and good diversity.

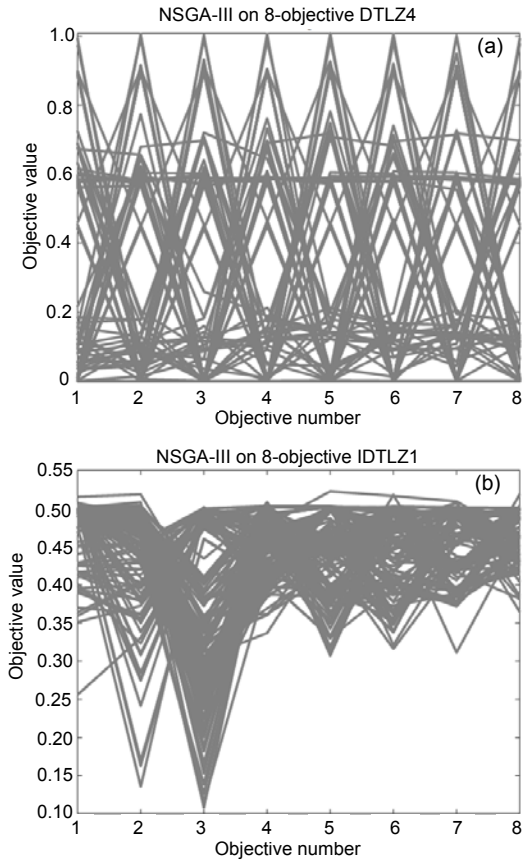

Fig. 1 Illustration of final populations obtained by NSGA-III on DTLZ4 (a) and IDTLZ1 (b) with eight objectives

Cai et al. (2015) proposed an EA-guided MOEA based on decomposition (EAG-MOEA/D). It uses an EA to guide the dynamic allocation of computational resources among subproblems. The method of dynamic resource allocation (DRA) can be used to decide which search regions should be searched at each generation. EAG-MOEA/D works well on combinatorial MOPs with two or three objectives.

Due to its ability of guiding the dynamic allocation of computational resources among subproblems, DRA has a distinctive advantage in solving MOPs with irregular PF. However, similar to other MOEAs, as the number of objective functions increases, EAG-MOEA/D also faces the loss of selection pressure and population diversity, which may deteriorate its performance for MaOPs with irregular PF.

In this study, an MaOEA based on decomposition with DRA (MaOEA/D-DRA) is proposed for irregular optimization. The contributions of this paper can be summarized as follows:

1. We find that although most existing MaOEAs can be used to solve MaOPs, they are considered mainly for problems of regular PF. In contrast to the problems with irregular PF, these algorithms find it difficult to reach satisfaction.

2. We introduce a new method of DRA to solve MaOPs with irregular PF (MaDRA), in which there are two populations during the evolutionary process, namely, evolutionary population (EP) and external archive (EA). EP uses the Tchebycheff approach to decompose an MaOP into several subproblems for evolution, and EA adopts a method of shift-based density estimation (SDE) (Li MQ et al., 2014) for updating, which enables the proposed MaOEA/D-DRA to solve MaOPs.

3. We conduct experiments to compare the proposed algorithm with five state-of-the-art MaOEAs using a variety of test problems with irregular PF; experimental results show that MaOEA/D-DRA outperforms the other algorithms in most test problems. To verify the effectiveness of the Tchebycheff approach in the decomposing process, experimental studies are conducted to compare the integration of the Tchebycheff approach into the proposed algorithm with two widely used decomposition approaches, namely, the weighted-sum approach and the penalty-based boundary intersection approach. Experimental results show that the Tchebycheff approach has better performance than these two decomposition approaches in terms of both convergence speed and diversity.

2 Related work

Decomposition is a well-known strategy in traditional multi-objective optimization. As a classical MOEA based on decomposition, MOEA/D decomposes an MOP into several single-objective subproblems with a scalarizing method; here, each solution corresponds to a weight vector (Zhang QF and Li, 2007), and all the subproblems are optimized in a collaborative manner. Each subproblem has an optimal solution with respect to a nondominated solution of MOP. During the evolutionary process, for each subproblem, two solutions are randomly selected as parents from its neighboring subproblems, and a new solution is generated using genetic operations. For the current subproblem and its neighboring subproblems, if the new solution performs better, the current solution is replaced with the new one, so a good solution can survive at multiple subproblems.

Since MOEA/D shows good performance in solving MOPs, decomposition-based MOEAs have attracted significant attention from researchers. Li H and Zhang (2009) followed the MOEA/D framework and proposed a new version of MOEA/D based on differential evolution (MOEA/D-DE), which is designed for MOPs with complicated Pareto set shapes. Zhang QF et al. (2010) presented an MOEA/D with the Gaussian stochastic process model, named MOEA/D-EGO, for dealing with expensive MOPs. Qi et al. (2014) have proposed an improved MOEA/D with adaptive weight vector adjustment (MOEA/D-AWA). Based on the geometric relationship between the weight vectors and the optimal solutions under the Tchebycheff approach, the weight vector can adaptively adjust in the evolution process. To solve MaOPs, Asafuddoula et al. (2015) proposed an improved decomposition-based EA, termed I-DBEA, where the balance between diversity and convergence speed is maintained using a simple preemptive distance comparison scheme. To solve MOPs and MaOPs with more flexibility, a new variant of MOEA/D with sorting and selection was presented by Cai et al. (2017) (MOEA/D-SAS), in which decomposition-based-sorting (DBS) and angle-based-selection (ABS) are used for the balance between convergence and diversity. To improve the diversity and reduce the sensitivity to the shapes of Pareto fronts, Cai et al. (2018) proposed a constrained decomposition with grids (CDG-MOEA), because the grids have the inherent property of reflecting the information of neighborhood structures of the solutions. Wang TC and Ting (2018) presented a fitness-inheritance-assisted MOEA/D using the covariance matrix adaptation evolution strategy, termed MOEA/D-FICMAES, for complex MOPs, where fitness inheritance is adopted to reduce the computational cost and information sharing facilitates communication and utilization of offspring information among different subproblems. To solve the constrained MOPs, Zhu et al. (2019) proposed a constrained version of MOEA/D with two types of weight vectors (MOEA/D-TW): the solutions associated with the convergence weight vectors are renewed according to only

the aggregation function, while the ones associated with diversity weight vectors are updated by considering both the aggregation function and the overall constraint violation.

EAG-MOEA/D is a variant of MOEA/D (Cai et al., 2015), designed for combinational optimization. In EAG-MOEA/D, DRA is proposed to guide the population for evolution. It maintains an evolutionary population EP and an external archive EA. EP adopts the weighted-sum approach to decompose an MOP into N single-objective optimization subproblems following the main framework of decomposition. EA is used to guide the dynamic allocation of computational resources among subproblems. Specifically, at the end of each generation k, EP is updated with N new solutions. Thereafter, the newly generated solutions are combined with EA, and the NSGA-II sorting approach is used to select the top N better solutions to update EA. We call a new solution successful if it enters EA, and record the number of successful solutions generated by subproblem i at each generation k as $s_{i,k}$, where G is the current generation, and L is the number of previous learning generations.

$$S_{i,G} = \sum_{k=G-L}^{G-1} s_{i,k}. \tag{3}$$

The probability of selecting subproblem i at each generation $G>L-1$ is defined as

$$p_{i,G} = D_{i,G} \Big/ \sum_{j=1}^{N} D_{j,G}, \tag{4}$$

where

$$D_{i,G} = S_{i,G} \Big/ \sum_{j=1}^{N} S_{j,G} + \varepsilon, \quad i = 1, 2, ..., N. \tag{5}$$

$D_{i,G}$ is the proportion of successful solutions generated by subproblem i over L. Here, ε=0.002 is used to ensure that all the $D_{i,G}$'s are larger than 0.

The framework of DRA is shown in Fig. 2; EP and EA complement each other. On one hand, with the help of EA, EP uses the decomposition approach for evolution. On the other hand, EA uses the new solutions generated from EP for updating. Therefore, EAG-MOEA/D can dynamically allocate computational resources to different subproblems.

There have been numerous studies on MOEAs based on decomposition for various test functions and real-world problems (Santiago et al., 2014; Trivedi et al., 2017), which constitute a research direction of great value.

3 The proposed algorithm

We propose an MaOEA based on decomposition with DRA for irregular optimization. In the proposed MaOEA/D-DRA, MaDRA is designed for solving MaOPs with irregular PF. Compared with DRA in EAG-MOEA/D, the main differences of MaDRA are the updating method of EA and the decomposition approach of EP.

3.1 Updating method of EA

EAG-MOEA/D uses DRA to guide the solutions to some special regions during the evolutionary process. EA adopts the NSGA-II sorting approach for updating, and it works well for MOPs with two or three objectives. Since MaOEA/D-DRA is focused on solving MaOPs that have more than three objectives, the NSGA-II sorting approach is not suitable due to a large number of nondominated solutions in EA.

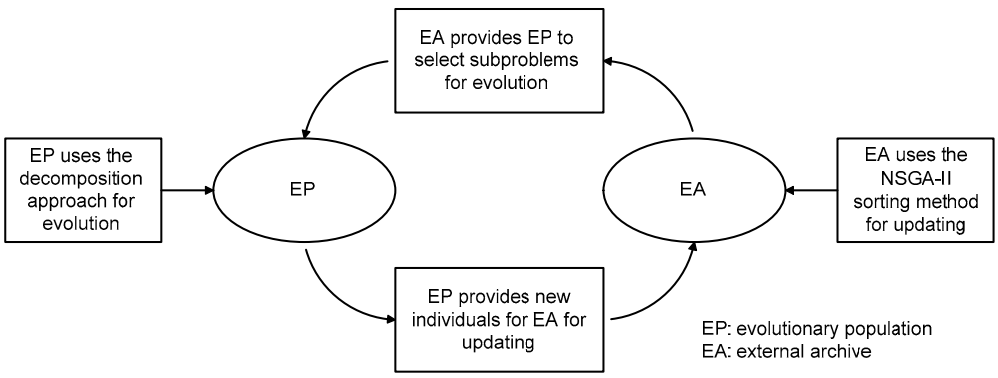

Fig. 2 Illustration of the framework of dynamic resource allocation (DRA)

Li MQ et al. (2014) proposed an SDE for Pareto-based algorithms in many-objective optimization. SDE considers both convergence speed and diversity for MaOPs, and it can be applied to any specific density estimator without additional parameters. Due to the robust performance and simple implementation of SDE, we use this method in MaOEA/D-DRA for MaOPs with irregular PF.

SDE estimates the convergence of the solution by adjusting the relative positions of other solutions in the population. The new density $D'(\boldsymbol{p}, P)$ of solution $\boldsymbol{p}$ in population P can be defined as follows:

$$D'(\boldsymbol{p}, P) = D\big(\mathrm{dist}(\boldsymbol{p}, \boldsymbol{q}_1'), \mathrm{dist}(\boldsymbol{p}, \boldsymbol{q}_2'), ..., \mathrm{dist}(\boldsymbol{p}, \boldsymbol{q}_{N-1}')\big), \tag{6}$$

where $\mathrm{dist}(\boldsymbol{p}, \boldsymbol{q}_i')$ denotes the similarity degree between solutions $\boldsymbol{p}$ and $\boldsymbol{q}_i'$, N is the size of P, and $\boldsymbol{q}_i'$ is the shifted version of solution $\boldsymbol{q}_i$ ($\boldsymbol{q}_i \in P$ and $\boldsymbol{q}_i \neq \boldsymbol{p}$), which is formulated as follows:

$$\boldsymbol{q}_{i(j)}' = \begin{cases} \boldsymbol{p}_{(j)}, & \text{if } \boldsymbol{q}_{i(j)} < \boldsymbol{p}_{i(j)}, \\ \boldsymbol{q}_{i(j)}, & \text{otherwise}, \end{cases} \quad j = 1, 2, ..., m, \tag{7}$$

where m is the number of objectives, and $\boldsymbol{p}_{(j)}$, $\boldsymbol{q}_{i(j)}$, and $\boldsymbol{q}'_{i(j)}$ denote the jth objective values of solutions $\boldsymbol{p}$, $\boldsymbol{q}_i$, and $\boldsymbol{q}_i'$, respectively.

In solving MaOPs with SDE, only solutions with both high convergence speed and good diversity have a low crowding degree. Solutions with either poor convergence speed or poor diversity have a high crowding degree. Consequently, SDE can reflect the convergence speed and diversity of the whole population during the evolutionary process.

3.2 Decomposition approach of EP

During the population evolutionary process, the optimal solution of the whole population in each generation plays an important role in speeding up the convergence of the algorithm. However, for simplicity, EAG-MOEA/D adopts the weighted-sum approach for decomposition, in which the optimal solution is not involved. In addition, EAG-MOEA/D is designed for solving discrete variable combinational optimization problems with two or three objectives, whereas MaOEA/D-DRA is designed for solving MaOPs with continuous variables. Therefore, in MaOEA/D-DRA, the weighted-sum approach is replaced by the Tchebycheff approach (Zhang QF and Li, 2007), which uses the optimal solution as a reference point to promote the algorithm's convergence performance. The Tchebycheff approach works as the following.

The original MOP is decomposed into N subproblems, which are optimized in a cooperative manner. Specifically, the objective function of the jth subproblem is

$$\begin{aligned} &\min\ g(\boldsymbol{x} \mid \boldsymbol{\lambda}^j, \boldsymbol{z}^*) = \max_{1 \le i \le m} \left(\lambda_i^j \left| f_i(\boldsymbol{x}) - z_i^* \right| \right) \\ &\text{s.t. } \boldsymbol{x} \in \Omega,\ j = 1, 2, ..., N, \end{aligned} \tag{8}$$

where $\{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N\}$ is a set of evenly spread weight vectors, $\boldsymbol{\lambda}^j = (\lambda_1^j, \lambda_2^j, ..., \lambda_m^j)^{\mathrm{T}}$, and $\boldsymbol{z}^* = (z_1^*, z_2^*, ..., z_m^*)$ is the reference point, i.e., $z_i^* = \min\{f_i(\boldsymbol{x}) \mid \boldsymbol{x} \in \Omega\}$ for each i=1, 2, …, m.

For each j=1, 2, …, N, let $B(j)$ be the neighborhood set of weight vector $\boldsymbol{\lambda}^j$, which contains $\boldsymbol{\lambda}^j$'s several closest weight vectors in $\{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N\}$ in terms of the Euclidean distance.

3.3 Framework of MaOEA/D-DRA

At each generation, MaOEA/D-DRA maintains the following:

1. an evolutionary population EP=$\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$, where $\boldsymbol{x}^i$ is the current solution to the ith subproblem,

2. $\boldsymbol{z}=(z_1, z_2, \ldots, z_m)^{\mathrm{T}}$, where z_i is the best solution found so far for objective i, and

3. an EA, which has N solutions selected by SDE.

MaOEA/D-DRA works as follows:

Step 1 (initialization): initialize EP, EA, and $\boldsymbol{z}$.

Step 2 (new solution generation): generate a set of N new solutions, Y, with genetic operators and MaDRA.

Step 3 (update): update EP, $\boldsymbol{z}$, and EA.

Step 4 (termination): if the maximum number of function evaluations is satisfied, output EA; otherwise, go to step 2.

In step 1, an initial population EP=$\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ is generated by randomly sampling from the decision variable space Ω. For simplicity, EA is initialized to be EP. The reference point $\boldsymbol{z}=(z_1, z_2, \ldots, z_m)^{\mathrm{T}}$ is initialized by setting $z_j=\min_{1\le i\le N} f_j(\boldsymbol{x}^i)$.

In step 2, according to MaDRA, a subproblem is selected based on the selection probability; then, two parent solutions are randomly selected from its neighbors and an offspring is generated with genetic operators. Note that a subproblem may be selected more than once at one generation.

In step 3, EP, $\boldsymbol{z}$, and EA are updated with Y. For EP, if the new solution $\boldsymbol{y}^j$ of subproblem i performs better than neighbors $\boldsymbol{x}^k$ with respect to the aggregated function of subproblem k, $\boldsymbol{x}^k$ will be replaced with $\boldsymbol{y}^j$. For $\boldsymbol{z}$, z_l is replaced with the value of the new solution $\boldsymbol{y}^j$ if $f_l(\boldsymbol{y}^j)$ performs better than z_l. Lastly, for EA, the combined population Z is obtained by merging EA and Y, and then the best N solutions are selected from Z to construct a new EA using SDE.

In step 4, if the maximum number of function evaluations is reached, the process stops and EA is outputted. Otherwise, go to step 2. Algorithm 1 gives the pseudocode.

Algorithm 1 MaOEA/D-DRA

Input: an MaOP with irregular PF; maximum number of function evaluations; number of subproblems, N; population size of EP and EA; a uniform spread of N weight vectors: $\boldsymbol{\lambda}^1$, $\boldsymbol{\lambda}^2$, …, $\boldsymbol{\lambda}^N$; number of weight vectors in the neighborhood of each subproblem, T.

Output: EA.

Step 1: initialization

Generate an initial population EP=$\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ by randomly sampling from Ω.

Set EA=EP.

Compute the Euclidean distance between any two of the weight vectors and work out the T closest weight vectors to each weight vector. For each i=1, 2, …, N, set $B(i)=\{i_1, i_2, \ldots, i_T\}$, where $\boldsymbol{\lambda}^{i_1}, \boldsymbol{\lambda}^{i_2}, \ldots, \boldsymbol{\lambda}^{i_T}$ are the T closest weight vectors to $\boldsymbol{\lambda}^i$.

Initialize $\boldsymbol{z}=(z_1, z_2, \ldots, z_m)^{\mathrm{T}}$ by setting $z_j=\min_{1\le i\le N} f_j(\boldsymbol{x}^i)$.

Step 2: new solution generation

For all $j\in\{1, 2, \ldots, N\}$ **do**

Select subproblem i for search according to Eq. (4).

Randomly select two indexes k and l from $B(i)$, and then generate a new solution $\boldsymbol{y}^j$ from $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$ for subproblem i by genetic operators.

End for

Step 3: update

For all $j\in\{1, 2, \ldots, N\}$ **do**

Update of neighboring solutions: if $\boldsymbol{y}^j$ is generated from subproblem i, for each index $k\in B(i)$, if $g(\boldsymbol{y}^j|\boldsymbol{\lambda}^k, \boldsymbol{z})\le g(\boldsymbol{x}^k|\boldsymbol{\lambda}^k, \boldsymbol{z})$, set $\boldsymbol{x}^k=\boldsymbol{y}^j$.

Update of $\boldsymbol{z}$: for each l=1, 2, …, m, if $f_l(\boldsymbol{y}^j)<z_l$, set $z_l=f_l(\boldsymbol{y}^j)$.

End for

Update of EA: obtain Z=EA$\cup Y$; select the best N solutions from Z to replace EA using SDE.

Step 4: termination

If the maximum number of function evaluations is reached, stop and output EA; otherwise, go to step 2.

3.4 Computational complexity of MaOEA/D-DRA

To analyze the computational complexity of MaOEA/D-DRA, the main steps in one generation in the main loop of Algorithm 1 are considered. Apart from the generation of the new solution in step 2, the main computational cost results from the initialization in step 1 and update in step 3.

As shown in Algorithm 1, the initialization of step 1 consists of two components: computing of the Euclidean distance and initialization of $\boldsymbol{z}$. The computational complexity of the computing of the Euclidean distance is $O(N^2)$, where N is the population size. The computational complexity of the initialization of $\boldsymbol{z}$ is $O(N\cdot m)$, where m is the objective number. In addition, the update of step 3 consists of three components: updating of neighboring solutions, updating of $\boldsymbol{z}$, and updating of EA. The computational complexity of updating of neighboring solutions is $O(N\cdot m\cdot T)$, where T is the number of the neighbor weight vectors of each subproblem. The computational complexities of $\boldsymbol{z}$ update and EA update are $O(N\cdot m)$ and $O(N^2\cdot m)$, respectively.

In conclusion, since $T\le N$, the overall computational complexity of MaOEA/D-DRA within one generation is $O(N^2\cdot m)$, which indicates that MaOEA/D-DRA is computationally efficient.

4 Experimental results and analysis

In this section, we compare MaOEA/D-DRA with five state-of-the-art MaOEAs, namely NSGA-III, MOEA/D, RVEA, MOEA/DD, and MaOEA-R&D. NSGA-III is a variant of NSGA-II, which is a classical Pareto dominance-based MOEA. MOEA/D, RVEA, and MaOEA-R&D are popular decomposition-based MaOEAs, whereas MOEA/DD is an MaOEA based on both dominance and decomposition. In addition, the three widely used decomposition approaches, namely the Tchebycheff approach, weighted-sum approach, and penalty-based boundary intersection approach, are compared in the same algorithmic framework.

4.1 Experimental settings

In the experiments, 11 unconstrained test problems with irregular PF and one constrained test problem from four widely used test suites, namely DTLZ5–DTLZ7 (Deb et al., 2005), IDTLZ1, IDTLZ2, C1_DTLZ1 (Jain and Deb, 2014), WFG1–WFG3 (Huband et al., 2006), MaF2, MaF4, and MaF13 (Cheng et al., 2018) have been used, as recommended by Tian et al. (2018); the relevant settings are given in Table 1. Each algorithm is run 30 times for each test problem independently, and the mean and standard deviation of each metric are recorded. The Wilcoxon rank-sum test with a significance level of 0.05 is used to compare the results obtained by MaOEA/D-DRA and the five compared algorithms, where the symbols "+," "–," and "≈" denote that the result obtained by another algorithm is significantly better, significantly worse, and statistically similar to that obtained by MaOEA/D-DRA, respectively. All the compared algorithms in this study are implemented on a MATLAB platform named PlatEMO (Tian et al., 2017). The details of the experimental settings are given in the following.

Reference points: As recommended by Deb and Jain (2014), uniformly distributed reference points with two layers are adopted in all the compared algorithms. Table 2 lists the number of reference points for varying numbers of objectives, where φ_1 and φ_2 are the numbers controlling the number of weight vectors along the boundary of the Pareto front and inside it, respectively. To ensure a fair comparison, the population size of each compared algorithm is set to the same as the number of reference points.

Genetic operators: In all the compared algorithms, simulated binary crossover (SBX) (Tian et al., 2018) and polynomial mutation (Deb and Goyal, 1996) are adopted to create offspring solutions. The distribution indexes of both SBX and polynomial mutation are set to 20. The crossover probability and mutation are set to 1.0 and $1/D$, respectively, where D denotes the number of decision variables.

Specific parameter setting: For NSGA-III and MaOEA-R&D, there is no additional parameter to be specified. For MOEA/D, the size of the neighborhood T is specified as 10% of the population size, and the Tchebycheff approach is selected as the aggregation function. For RVEA, the penalty parameter α and the frequency of reference point adaption f_r are set to 2 and 0.1, respectively. For MOEA/DD, T is set to the same value as in MOEA/D, and the neighborhood selection probability δ is set to 0.9. For the proposed MaOEA/D-DRA, T is set to the same value as in MOEA/D and MOEA/DD, and L is set to 8.

Table 2 Settings of the number of reference points for each number of objectives

Number of objectives	(φ_1, φ_2)	Number of reference points
5	(4, 3)	105
8	(3, 2)	156
10	(3, 1)	230

φ_1 and φ_2 are the numbers of weight vectors along the boundary of the Pareto front and inside it, respectively

Table 1 Settings of the number of objectives, the number of decision variables, the maximum number of function evaluations, and the features of Pareto front for each test problem

Problem	M	D	E	Pareto front
DTLZ5, DTLZ6	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Mostly degenerate
DTLZ7	5, 8, 10	$M-1+20$	21 000, 31 200, 46 000	Disconnected
IDTLZ1	5, 8, 10	$M-1+5$	52 500, 78 000, 11 5000	Inverted
IDTLZ2	5, 8, 10	$M-1+10$	21 000, 31 200, 46 000	Inverted
C1_DTLZ1	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Constrained problem
WFG1	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Sharp tails
WFG2	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Disconnected
WFG3	5, 8, 10	$M-1+10$	21 000, 31 200, 46 000	Mostly degenerate
MaF2	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Disconnected
MaF4	5, 8, 10	$M-1+10$	52 500, 78 000, 11 5000	Inverted
MaF13	5, 8, 10	5	52 500, 78 000, 11 5000	Degenerate

M: number of objectives; D: number of decision variables; E: maximum number of function evaluations

4.2 Performance metrics

To make an empirical comparison between the solution sets of different algorithms, two widely used performance metrics are adopted to measure the solution sets in terms of both convergence speed and diversity.

The first performance indicator used is the inverted generational distance (IGD) (Zhou et al., 2006). It measures the average distance from a set of evenly distributed reference points P^* in PF to the set of nondominated solutions Ω found by MaOEAs. It can be formulated as follows:

$$\mathrm{IGD}(P^*,\Omega)=\frac{\sum_{x\in P^*}\mathrm{dis}(\boldsymbol{x},\Omega)}{|P^*|}, \tag{9}$$

where dis($\boldsymbol{x}$, Ω) is the minimum Euclidean distance between $\boldsymbol{x}$ and points in Ω, and $|P^*|$ denotes the size of P^*. For the experiments in this study, roughly 10 000 uniformly distributed points are selected from the PF to form P^* by Das and Dennis (1998)'s approach. The set Ω with smaller IGD values is better.

The second performance indicator applied in this study is the hypervolume (HV) (He et al., 2017). In contrast to IGD, larger HV values mean better quality. Generally speaking, the HV value of a solution set Ω is formulated as the area covered by Ω with regard to a set of predefined reference points P^* in the objective space:

$$\mathrm{HV}(\Omega, P^*)=\lambda\big(H(\Omega, P^*)\big), \tag{10}$$

where

$$H(\Omega, P^*)=\big\{\boldsymbol{z}\in Z\,|\,\exists\boldsymbol{x}\in P,\,\boldsymbol{r}\in P^*:\ f(\boldsymbol{x})_i\le z_i\le r_i,\ i=1,2,\ldots,m\big\}, \tag{11}$$

and λ is the Lebesgue measure with

$$\lambda\big(H(\Omega, P^*)\big)=\int_{P^{*m}}1_{H(\Omega, P^*)}(z)\mathrm{d}z, \tag{12}$$

where $1_{H(\Omega, P^*)}$ is the characteristic function of $H(\Omega, P^*)$.

4.3 Comparisons of MaOPs with irregular PF

Table 3 gives the IGD values obtained by the six algorithms over 30 independent runs for the 36 instances of 12 test problems, with the best results being highlighted. In general, MaOEA/D-DRA significantly outperforms the five other algorithms. On one hand, MaOEA/D-DRA shows the best performance in 20 of 36 instances, with the best results obtained by NSGA-III, MOEA/D, RVEA, MOEA/DD, and MaOEA-R&D being 3, 7, 0, 3, and 3, respectively. On the other hand, compared with all the five other algorithms, MaOEA/D-DRA has achieved more than 25 significantly better results.

Table 4 summarizes the statistical results of the HV values obtained by the six algorithms over 30 independent runs for the 36 instances of the 12 test problems, where the best results are highlighted. In general, MaOEA/D-DRA has achieved the best performance in 16 of 36 instances, and the best results obtained by NSGA-III, MOEA/D, RVEA, MOEA/DD, and MaOEA-R&D are 8, 7, 4, 1, and 0, respectively. Compared with NSGA-III, MOEA/D, RVEA, MOEA/DD, and MaOEA-R&D, the numbers of significantly better results obtained by MaOEA/D-DRA are 20, 26, 26, 24, and 31, respectively, while the numbers of significantly worse results obtained by MaOEA/D-DRA are only 12, 6, 7, 9, and 2, respectively.

From the above discussions, MaOEA/D-DRA significantly outperforms all the five other algorithms in terms of both convergence speed and diversity. This is because MaDRA can guide the solutions to some special regions according to the shape of the irregular PF during the evolutionary process.

Note that, for constrained problem C1_DTLZ1, MaOEA/D-DRA does not obtain the optimal value in all the three test instances. In terms of the IGD value, NSGA-III, RVEA, and MOEA/DD perform better than MaOEA/D-DRA in all the three instances. Compared with MOEA/D, MaOEA/D-DRA obtains significantly better results in all the three test instances. Compared with MaOEA-R&D, MaOEA/D-DRA obtains two significantly better results and one statistically similar result. In terms of the HV value, RVEA and MOEA/DD perform better than MaOEA/D-DRA in all the three instances. Compared with NSGA-III, MaOEA/D-DRA obtains two significantly worse results and one statistically similar result. Compared with MOEA/D and MaOEA-R&D,

Table 3 IGD results of MaOEA/D-DRA and five other algorithms in 36 test instances

Problem	M	IGD					
		NSGA-III	MOEA/D	RVEA	MOEA/DD	MaOEA-R&D	MaOEA/D-DRA
DTLZ5	5	1.5355e–1	**3.4676e–2**	4.0564e–1	9.9241e–2	7.4209e–1	6.7229e–2
		(7.20e–2) –	**(8.99e–4) +**	(5.93e–2) –	(1.34e–2) –	(2.66e–8) –	(8.25e–3)
	8	2.4557e–1	**4.3631e–2**	3.8414e–1	1.5326e–1	7.4205e–1	1.3701e–1
		(8.06e–2) –	**(3.84e–3) +**	(5.44e–2) –	(1.85e–2) –	(2.12e–4) –	(3.05e–2)
	10	3.1546e–1	**4.8327e–2**	2.3049e–1	1.2067e–1	7.4209e–1	1.1945e–1
		(7.33e–2) –	**(4.25e–3) +**	(2.17e–2) –	(5.77e–3) ≈	(3.25e–8) –	(1.75e–2)
DTLZ6	5	3.3480e–1	**3.2975e–2**	2.6525e–1	1.0130e–1	8.9463e–1	9.0757e–2
		(1.82e–1) –	**(2.43e–3) +**	(5.00e–2) –	(1.13e–2) –	(9.12e–1) –	(1.57e–2)
	8	5.9784e–1	**3.5465e–2**	2.9097e–1	1.6927e–1	8.6616e–1	1.4337e–1
		(3.19e–1) –	**(5.15e–3) +**	(5.06e–2) –	(2.79e–2) –	(1.27e+0) –	(2.99e–2)
	10	2.8277e+0	**4.4143e–2**	1.7661e–1	1.1576e–1	7.2180e–1	1.2049e–1
		(1.36e+0) –	**(9.44e–4) +**	(3.40e–2) –	(6.45e–4) ≈	(7.11e–2) –	(1.80e–2)
DTLZ7	5	3.4488e–1	6.1481e–1	4.7679e–1	2.9096e+0	1.0536e+0	**3.2191e–1**
		(2.67e–2) –	(6.62e–2) –	(2.42e–2) –	(3.47e–1) –	(4.92e–1) –	**(4.32e–2)**
	8	9.3899e–1	1.1122e+0	1.3464e+0	1.6944e+0	1.4547e+0	**7.6641e–1**
		(8.88e–2) –	(2.29e–1) –	(2.73e–1) –	(4.80e–1) –	(2.87e–1) –	**(3.40e–2)**
	10	1.3719e+0	1.6324e+0	2.0265e+0	2.2232e+0	1.9864e+0	**1.0055e+0**
		(1.96e–1) –	(5.92e–1) –	(6.33e–1) –	(4.54e–1) –	(3.54e–1) –	**(1.09e–1)**
IDTLZ1	5	1.0322e–1	7.6672e–2	1.8852e–1	1.4780e–1	2.3172e–1	**7.0119e–2**
		(8.12e–3) –	(1.59e–3) –	(2.13e–2) –	(5.76e–3) –	(2.40e–2) –	**(7.83e–4)**
	8	1.3131e–1	1.2024e–1	2.5781e–1	2.1602e–1	2.8432e–1	**1.1130e–1**
		(6.98e–3) –	(8.92e–4) –	(2.48e–2) –	(1.25e–2) –	(2.56e–2) –	**(8.20e–4)**
	10	1.4219e–1	1.4130e–1	2.7651e–1	1.8161e–1	2.8721e–1	**1.2592e–1**
		(3.11e–3) –	(2.57e–3) –	(4.23e–2) –	(1.42e–2) –	(2.20e–2) –	**(4.15e–3)**
IDTLZ2	5	2.5657e–1	2.2640e–1	3.2925e–1	2.5239e–1	6.8831e–1	**2.1889e–1**
		(1.56e–2) –	(1.39e–3) –	(2.13e–2) –	(7.99e–3) –	(5.32e–3) –	**(4.87e–3)**
	8	4.7460e–1	**3.4820e–1**	6.1128e–1	6.4481e–1	8.4950e–1	3.6913e–1
		(1.40e–2) –	**(1.21e–3) +**	(5.03e–3) –	(1.59e–2) –	(2.22e–2) –	(4.36e–3)
	10	5.7976e–1	4.4032e–1	6.6866e–1	6.8035e–1	8.7233e–1	**4.2794e–1**
		(1.28e–2) –	(3.26e–3) –	(4.15e–3) –	(5.22e–3) –	(2.33e–2) –	**(4.01e–3)**
C1_DTLZ1	5	7.1629e–2	1.1496e–1	8.2626e–2	**6.7300e–2**	2.5223e–1	9.3873e–2
		(1.25e–2) +	(1.88e–2) –	(2.64e–2) +	**(1.06e–2) +**	(6.48e–2) –	(2.23e–2)
	8	1.0773e–1	1.7782e–1	1.1274e–1	**9.3142e–2**	2.6794e–1	1.4100e–1
		(1.84e–2) +	(1.50e–2) –	(2.24e–2) +	**(5.07e–3) +**	(0.00e+0) ≈	(1.54e–2)
	10	1.2468e–1	2.0402e–1	1.2834e–1	**1.1998e–1**	2.9173e–1	1.7481e–1
		(9.02e–3) +	(1.88e–2) –	(7.64e–3) +	**(7.20e–3) +**	(3.72e–2) –	(2.44e–2)
WFG1	5	7.2693e–1	1.5733e+0	6.5620e–1	1.3809e+0	1.4961e+0	**5.9809e–1**
		(7.34e–2) –	(7.06e–2) –	(8.94e–2) –	(1.54e–1) –	(2.76e–1) –	**(5.72e–2)**
	8	**1.0710e+0**	2.1841e+0	1.1600e+0	1.8228e+0	2.1933e+0	1.0789e+0
		(7.11e–2) ≈	(1.18e–1) –	(1.05e–1) –	(2.16e–1) –	(4.50e–1) –	(1.01e–1)
	10	**1.2235e+0**	2.5327e+0	1.3886e+0	1.6030e+0	2.3942e+0	1.5928e+0
		(6.24e–2) +	(2.01e–1) –	(6.98e–2) +	(1.64e–1) ≈	(6.04e–1) –	(3.93e–1)

To be continued

Table 3

Problem	M	IGD					
		NSGA-III	MOEA/D	RVEA	MOEA/DD	MaOEA-R&D	MaOEA/D-DRA
WFG2	5	8.0846e–1 (3.73e–2) +	1.7030e+0 (1.01e–1) –	1.9206e+0 (4.91e–1) –	4.0022e+0 (3.50e–1) –	**7.3105e–1 (6.53e–2) +**	1.2829e+0 (1.46e–1)
	8	2.0962e+0 (1.34e+0) +	3.2781e+0 (1.05e+0) –	4.2415e+0 (7.39e–1) –	7.4947e+0 (5.23e–2) –	**1.6293e+0 (1.67e–1) +**	2.5180e+0 (5.79e–1)
	10	5.1014e+0 (2.22e+0) –	4.3140e+0 (2.69e+0) –	8.4774e+0 (2.14e+0) –	1.4897e+1 (3.44e–1) –	**2.1743e+0 (2.43e–1) +**	4.1714e+0 (1.03e+0)
WFG3	5	7.2221e–1 (1.17e–1) –	1.4402e+0 (5.99e–2) –	7.7959e–1 (2.39e–1) –	9.4212e–1 (5.46e–2) –	1.8679e+0 (8.97e–1) –	**5.3937e–1 (7.17e–2)**
	8	1.0728e+0 (2.98e–1) –	2.5978e+0 (8.40e–2) –	2.4303e+0 (7.83e–1) –	2.0496e+0 (6.69e–2) –	2.1729e+0 (5.47e–1) –	**7.4950e–1 (1.08e–1)**
	10	**8.5996e–1 (1.33e–1) ≈**	3.2714e+0 (4.76e–1) –	4.0896e+0 (1.10e+0) –	2.7644e+0 (8.56e–2) –	2.8920e+0 (8.17e–1) –	8.6920e–1 (1.22e–1)
MaF2	5	1.3040e–1 (3.00e–3) –	1.3503e–1 (1.00e–3) –	1.2652e–1 (1.50e–3) –	1.4116e–1 (1.62e–2) –	4.9489e–1 (3.97e–2) –	**1.1677e–1 (2.05e–3)**
	8	2.2543e–1 (4.54e–2) –	2.7974e–1 (1.44e–2) –	2.4236e–1 (8.14e–2) –	2.0562e–1 (9.96e–3) –	7.9523e–1 (9.55e–2) –	**1.6103e–1 (3.08e–3)**
	10	2.2442e–1 (2.74e–2) –	3.1540e–1 (4.62e–2) –	3.6527e–1 (1.81e–1) –	2.3384e–1 (3.98e–2) –	8.2642e–1 (4.08e–2) –	**1.7244e–1 (3.27e–3)**
MaF4	5	5.3251e+0 (9.36e+0) –	4.3805e+0 (1.79e–1) –	5.6736e+0 (1.55e+0) –	7.0828e+0 (7.09e–1) –	1.6111e+1 (2.00e+0) –	**2.7955e+0 (1.65e–1)**
	8	2.9783e+1 (2.03e+0) –	3.2395e+1 (1.23e+0) –	5.7492e+1 (2.12e+1) –	9.5144e+1 (4.86e+0) –	1.4507e+2 (2.51e+1) –	**2.1480e+1 (1.30e+0)**
	10	1.0512e+2 (6.12e+0) –	1.3365e+2 (5.60e+0) –	1.9735e+2 (4.40e+1) –	4.0066e+2 (1.68e+1) –	5.0274e+2 (1.17e+2) –	**7.8675e+1 (3.34e+0)**
MaF13	5	2.3862e–1 (2.59e–2) –	1.8886e–1 (2.56e–2) –	6.2602e–1 (9.80e–2) –	2.3012e–1 (2.48e–2) –	9.1158e–1 (1.60e–1) –	**1.3036e–1 (1.43e–2)**
	8	2.5845e–1 (3.28e–2) –	2.5187e–1 (2.58e–2) –	7.2472e–1 (1.94e–1) –	3.8653e–1 (3.28e–2) –	1.0503e+0 (3.29e–1) –	**1.3739e–1 (2.38e–2)**
	10	2.5174e–1 (2.27e–2) –	3.6981e–1 (4.66e–2) –	9.1294e–1 (2.60e–1) –	2.9730e–1 (2.97e–2) –	1.2567e+0 (3.26e–1) –	**1.3345e–1 (1.24e–2)**
+/–/≈		6/28/2	7/29/0	4/32/0	3/30/3	3/32/1	

M: number of objectives. The best result in each row is highlighted in bold

MaOEA/D-DRA obtains two significantly better results and one statistically similar result. Therefore, MaOEA/D-DRA does not perform well in the constrained problem C1_DTLZ1. In terms of constraint optimization, MaOEA/D-DRA still needs improvement.

To intuitively compare the performance of the algorithms, we select three algorithms with different shapes of the irregular PF with 5, 8, and 10 objectives, namely 5-objective IDTLZ1, 8-objective DTLZ7, and 10-objective MaF4. Figs. 3–5 plot the parallel coordinates of the nondominated fronts on 5-objective IDTLZ1, 8-objective DTLZ7, and 10-objective MaF4, obtained by each algorithm with the median IGD value in 30 runs. It is evident from these figures that MaOEA/D-DRA can better approach the true irregular PF than all the five other algorithms.

Table 4 HV results of MaOEA/D-DRA and five other algorithms in 36 test instances

Problem	*M*	HV					
		NSGA-III	MOEA/D	RVEA	MOEA/DD	MaOEA-R&D	MaOEA/D-DRA
DTLZ5	5	6.0772e–3 (1.75e–3) –	**8.6141e–3** **(2.71e–5) +**	6.4779e–3 (1.91e–5) –	7.6871e–3 (2.07e–4) –	6.4705e–3 (4.64e–9) –	8.0417e–3 (1.62e–4)
	8	1.7190e–5 (5.65e–7) +	**1.8882e–5** **(4.88e–8) +**	1.6817e–5 (4.13e–9) –	1.7787e–5 (1.11e–7) +	1.6821e–5 (3.58e–11) –	1.7100e–5 (2.14e–7)
	10	3.4705e–8 (2.12e–8) –	**6.0378e–8** **(1.99e–10) +**	5.5708e–8 (5.01e–10) –	5.8209e–8 (2.04e–10) +	5.6218e–8 (5.08e–14) –	5.6548e–8 (2.49e–10)
DTLZ6	5	4.7538e–3 (2.84e–3) –	**8.6268e–3** **(2.51e–5) +**	6.7674e–3 (3.94e–4) –	7.7310e–3 (1.57e–4) ≈	5.4048e–3 (2.19e–3) –	7.7973e–3 (2.59e–4)
	8	9.2253e–6 (8.10e–6) –	**1.8943e–5** **(6.26e–8) +**	1.6649e–5 (3.06e–6) ≈	1.7725e–5 (4.23e–8) +	1.0313e–5 (8.13e–6) –	1.7025e–5 (1.64e–7)
	10	1.8739e–9 (1.03e–8) –	**6.0327e–8** **(1.98e–10) +**	5.5834e–8 (5.25e–9) ≈	5.8258e–8 (1.47e–10) +	5.1783e–8 (1.44e–8) –	5.6530e–8 (2.46e–10)
DTLZ7	5	1.9128e+0 (6.88e–2) ≈	1.4795e+0 (1.68e–1) –	1.7082e+0 (1.28e–1) –	7.9961e–1 (3.90e–3) –	1.4692e+0 (9.89e–2) –	**1.9345e+0** **(7.84e–2)**
	8	1.6172e+0 (1.94e–1) +	3.9241e–1 (4.96e–1) –	**1.6423e+0** **(2.08e–1) +**	2.9310e–1 (3.40e–1) ≈	5.4074e–1 (1.60e–1) +	4.7545e–1 (2.17e–1)
	10	**1.7749e+0** **(9.83e–2) +**	1.0321e–1 (1.20e–1) –	1.5293e+0 (1.74e–1) +	1.6093e–2 (8.15e–2) –	1.3884e–1 (7.20e–2) ≈	2.2034e–1 (2.48e–1)
IDTLZ1	5	2.4744e–4 (6.22e–5) –	4.0466e–4 (1.82e–5) –	7.2734e–5 (2.90e–5) –	1.1355e–4 (1.49e–5) –	5.1261e–5 (1.17e–5) –	**4.6325e–4** **(1.66e–5)**
	8	**1.9669e–7** **(3.63e–8) +**	1.2832e–7 (3.68e–8) –	1.1613e–8 (5.69e–9) –	3.9065e–8 (1.02e–8) –	7.8311e–9 (3.08e–9) –	1.6440e–7 (2.83e–8)
	10	**1.2892e–9** **(6.72e–11) +**	6.3786e–10 (1.55e–9) ≈	3.4615e–11 (2.74e–11) ≈	3.4811e–10 (8.42e–11) ≈	1.9658e–11 (7.47e–12) ≈	7.8369e–10 (1.20e–9)
IDTLZ2	5	9.6717e–2 (6.56e–3) –	9.7968e–2 (2.45e–3) –	1.2519e–1 (1.26e–2) –	1.2109e–1 (8.48e–3) –	4.5015e–2 (2.07e–3) –	**1.7200e–1** **(3.73e–3)**
	8	3.3012e–3 (7.35e–4) –	4.3478e–4 (4.99e–5) –	3.1234e–3 (2.93e–4) –	2.8098e–3 (4.26e–4) –	1.3516e–3 (4.01e–4) –	**5.1181e–3** **(5.20e–4)**
	10	**5.7321e–4** **(4.57e–5) +**	7.2393e–6 (3.77e–6) –	5.1337e–4 (3.53e–5) +	5.6072e–4 (2.46e–5) +	7.2551e–5 (3.19e–5) –	2.3708e–4 (6.48e–5)
C1_DTLZ1	5	4.3435e–2 (2.81e–3) ≈	4.1329e–2 (2.96e–3) ≈	**4.6356e–2** **(1.65e–3) +**	4.4491e–2 (2.56e–3) +	2.2888e–2 (7.61e–3) –	4.2046e–2 (3.42e–3)
	8	7.5565e–3 (5.93e–4) +	7.1082e–3 (3.19e–4) –	**8.0786e–3** **(1.21e–4) +**	7.7700e–3 (2.39e–4) +	4.9780e–3 (0.00e+0) ≈	7.3339e–3 (4.52e–4)
	10	2.3521e–3 (1.05e–4) +	1.9862e–3 (1.19e–4) –	**2.4800e–3** **(2.54e–5) +**	2.4190e–3 (5.80e–5) +	1.2138e–3 (2.84e–4) –	2.3034e–3 (7.20e–5)
WFG1	5	4.8129e+3 (2.62e+2) –	5.9884e+3 (1.08e+1) –	4.9659e+3 (2.83e+2) –	3.3045e+3 (3.76e+2) –	2.8856e+3 (7.61e+2) –	**6.0270e+3** **(3.32e+0)**
	8	1.6981e+7 (1.06e+6) –	2.0544e+7 (5.97e+5) –	1.8633e+7 (1.36e+6) –	1.3427e+7 (1.68e+6) –	8.4707e+6 (3.49e+6) –	**2.0663e+7** **(3.42e+5)**
	10	7.5787e+9 (3.96e+8) ≈	6.5019e+9 (1.21e+9) –	8.3719e+9 (4.96e+8) +	**8.4044e+9** **(3.61e+8) +**	3.6322e+9 (1.60e+9) –	7.6010e+9 (9.31e+8)
WFG2	5	**6.1285e+3** **(9.07e+0) +**	5.9563e+3 (6.61e+1) –	6.0262e+3 (3.30e+1) –	5.9933e+3 (4.33e+1) –	5.9043e+3 (5.28e+1) –	6.1186e+3 (1.73e+1)
	8	2.1978e+7 (4.72e+4) –	2.1936e+7 (7.44e+4) –	2.1663e+7 (1.01e+5) –	2.1153e+7 (1.35e+5) –	2.1149e+7 (5.45e+5) –	**2.2024e+7** **(3.92e+4)**
	10	9.6034e+9 (1.76e+7) ≈	9.5054e+9 (7.56e+7) –	9.4851e+9 (4.54e+7) –	9.2114e+9 (9.79e+7) –	9.3028e+9 (2.12e+8) –	**9.6043e+9** **(1.41e+7)**

To be continued

Table 4

Problem	M	HV					
		NSGA-III	MOEA/D	RVEA	MOEA/DD	MaOEA-R&D	MaOEA/D-DRA
WFG3	5	1.0486e+0 (3.07e–1) –	9.9211e–1 (2.26e–1) –	8.0637e–1 (3.73e–1) –	6.5164e–1 (2.69e–1) –	2.8132e–1 (5.57e–1) –	**1.5790e+0 (2.87e–1)**
	8	2.4794e–4 (7.34e–4) –	**1.0001e–2 (3.60e–3) ≈**	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	9.3936e–3 (6.33e–3)
	10	0.0000e+0 (0.00e+0) –	1.9200e–5 (1.15e–5) ≈	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	0.0000e+0 (0.00e+0) –	**2.7001e–5 (1.67e–5)**
MaF2	5	4.1962e–2 (1.27e–3) –	4.5000e–2 (2.09e–4) –	4.0892e–2 (5.04e–4) –	3.6084e–2 (2.25e–3) –	1.0043e–2 (1.14e–3) –	**5.0573e–2 (3.58e–4)**
	8	2.7468e–2 (6.94e–4) –	2.5983e–2 (2.89e–4) –	2.2292e–2 (2.24e–3) –	2.3948e–2 (7.13e–4) –	6.2090e–3 (2.51e–3) –	**2.9989e–2 (4.11e–4)**
	10	7.6925e–3 (2.19e–4) –	6.5276e–3 (1.16e–4) –	5.7821e–3 (1.58e–3) –	6.3221e–3 (3.40e–4) –	1.5866e–3 (2.29e–4) –	**8.0826e–3 (8.82e–5)**
MaF4	5	**2.5952e+3 (9.71e+2) +**	1.2499e+3 (2.46e+2) –	3.5709e+2 (2.02e+2) –	1.2148e+3 (4.10e+2) –	3.4058e+2 (3.67e+2) –	2.2155e+3 (5.79e+2)
	8	**2.7851e+8 (6.85e+7) +**	5.0962e+6 (1.79e+6) –	1.4568e+6 (7.99e+5) –	2.0693e+6 (6.75e+5) –	1.6004e+7 (2.45e+7) –	2.6243e+7 (1.52e+7)
	10	**1.6948e+13 (1.07e+12) +**	1.2906e+10 (2.94e+10) –	9.3379e+9 (6.71e+9) –	1.3915e+10 (3.76e+9) –	3.6295e+12 (5.01e+12) +	1.0429e+11 (4.90e+10)
MaF13	5	3.2014e–1 (2.16e–2) –	3.7407e–1 (1.77e–2) –	2.1396e–1 (7.02e–2) –	3.1758e–1 (3.68e–2) –	1.5451e–1 (2.49e–2) –	**4.2545e–1 (9.70e–3)**
	8	2.7436e–1 (2.30e–2) –	2.8887e–1 (1.33e–2) –	2.0022e–1 (5.38e–2) –	1.4587e–1 (3.38e–2) –	1.3083e–1 (5.39e–2) –	**3.5881e–1 (7.37e–3)**
	10	2.2438e–1 (5.94e–2) –	2.6754e–1 (9.08e–3) –	2.2781e–1 (5.30e–2) –	2.4270e–1 (1.96e–2) –	1.3745e–1 (6.87e–2) –	**3.5999e–1 (4.94e–3)**
+/–/≈		12/20/4	6/26/4	7/26/3	9/24/3	2/31/3	

M: number of objectives. The best result in each row is highlighted in bold

4.4 Impact of different decomposition approaches

In this subsection, we compare the Tchebycheff approach with the weighted-sum and penalty-based boundary intersection approaches. For fair comparisons, the Tchebycheff approach is replaced by the weighted-sum approach and penalty-based boundary intersection approach only in the decomposition process, while the other parts of MaOEA/D-DRA are retained. For simplicity, MaOEA/D-DRA with the weighted-sum approach is hereafter denoted as WS-MaOEA/D-DRA, MaOEA/D-DRA with the penalty-based boundary intersection approach is hereafter denoted as PBI-MaOEA/D-DRA. Tables 5 and 6 show the IGD and HV values, respectively, obtained by MaOEA/D-DRA, WS-MaOEA/D-DRA, and PBI-MaOEA/D-DRA over 30 independent runs in the 36 instances of the 12 test problems, where the best results are highlighted.

For the IGD value, MaOEA/D-DRA shows the best performance in 21 of the 36 instances, while the best result obtained by WS-MaOEA/D-DRA is only 8, and the best result obtained by PBI-MaOEA/D-DRA is only 7. Compared with WS-MaOEA/D-DRA and PBI-MaOEA/D-DRA, the numbers of significantly better results obtained by MaOEA/D-DRA are 25 and 27, respectively, while the number of significantly worse results obtained by MaOEA/D-DRA is 6 for both.

For the HV value, MaOEA/D-DRA has achieved the best performance in 14 of 36 instances, while the best results obtained by WS-MaOEA/D-DRA and PBI-MaOEA/D-DRA are 12 and 10, respectively. Compared with WS-MaOEA/D-DRA and PBI-MaOEA/D-DRA, the numbers of significantly better results obtained by MaOEA/D-DRA are 18 and 24, respectively, while the numbers of significantly worse

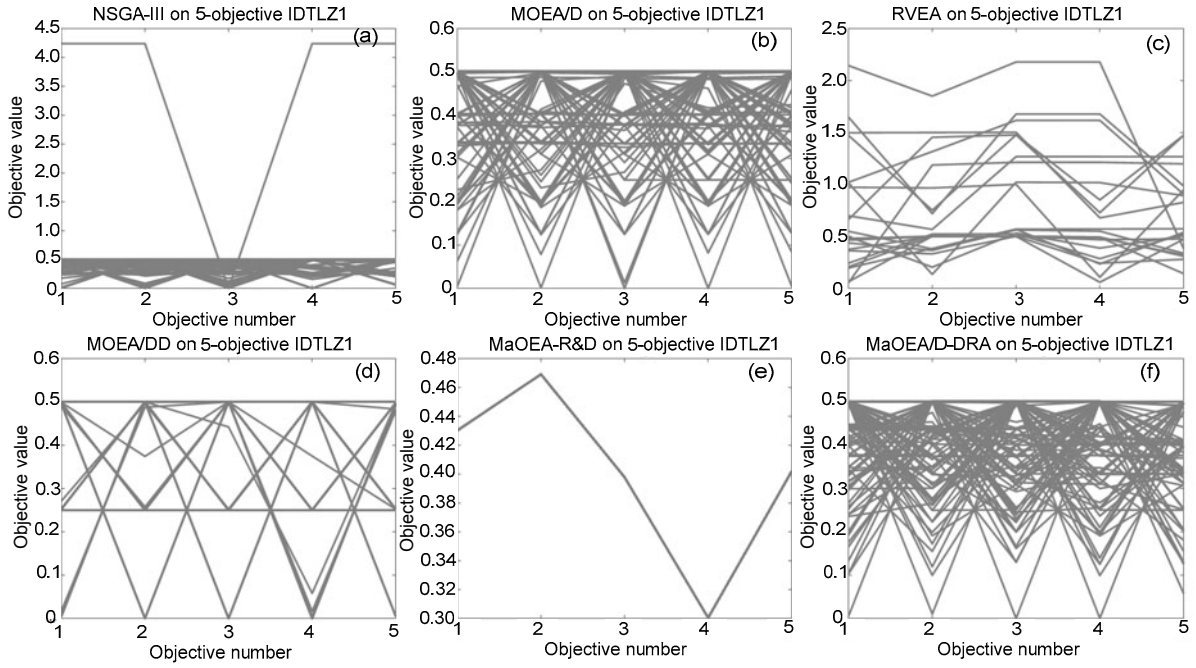

Fig. 3 Illustration of parallel coordinates of the non-dominated fronts on 5-objective IDTLZ1, obtained by NSGA-III (a), MOEA/D (b), RVEA (c), MOEA/DD (d), MaOEA-R&D (e), and MaOEA/D-DRA (f) with the median IGD value in 30 runs

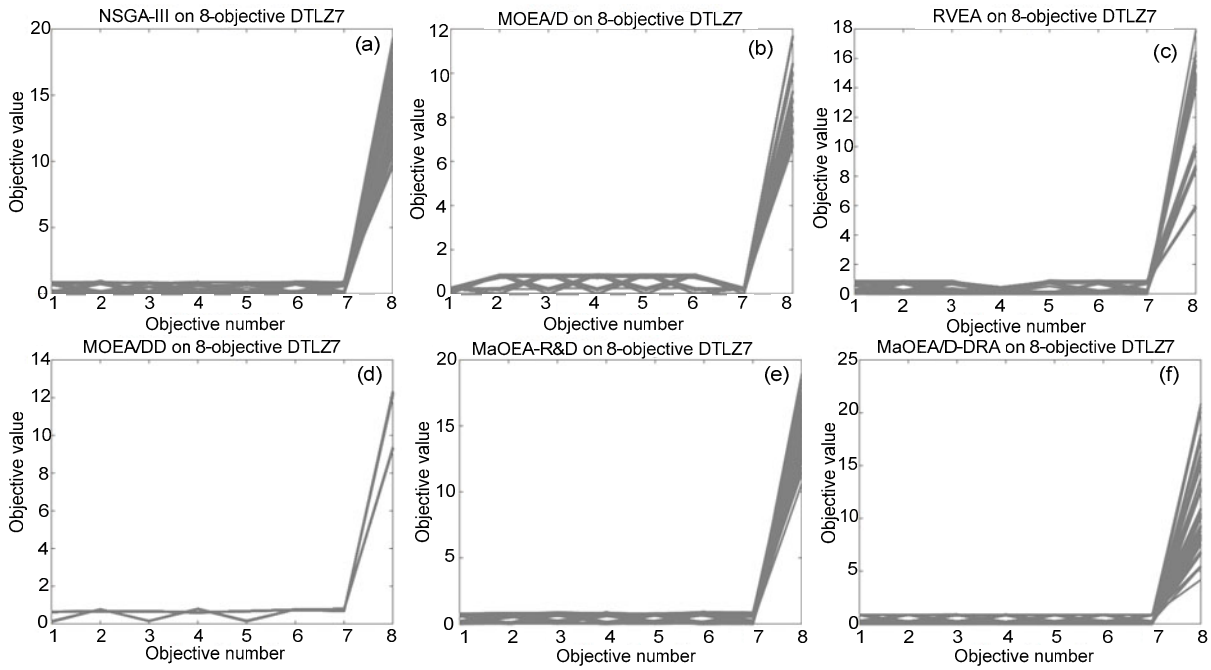

Fig. 4 Illustration of parallel coordinates of the non-dominated fronts on 8-objective DTLZ7, obtained by NSGA-III (a), MOEA/D (b), RVEA (c), MOEA/DD (d), MaOEA-R&D (e), and MaOEA/D-DRA (f) with the median IGD value in 30 runs

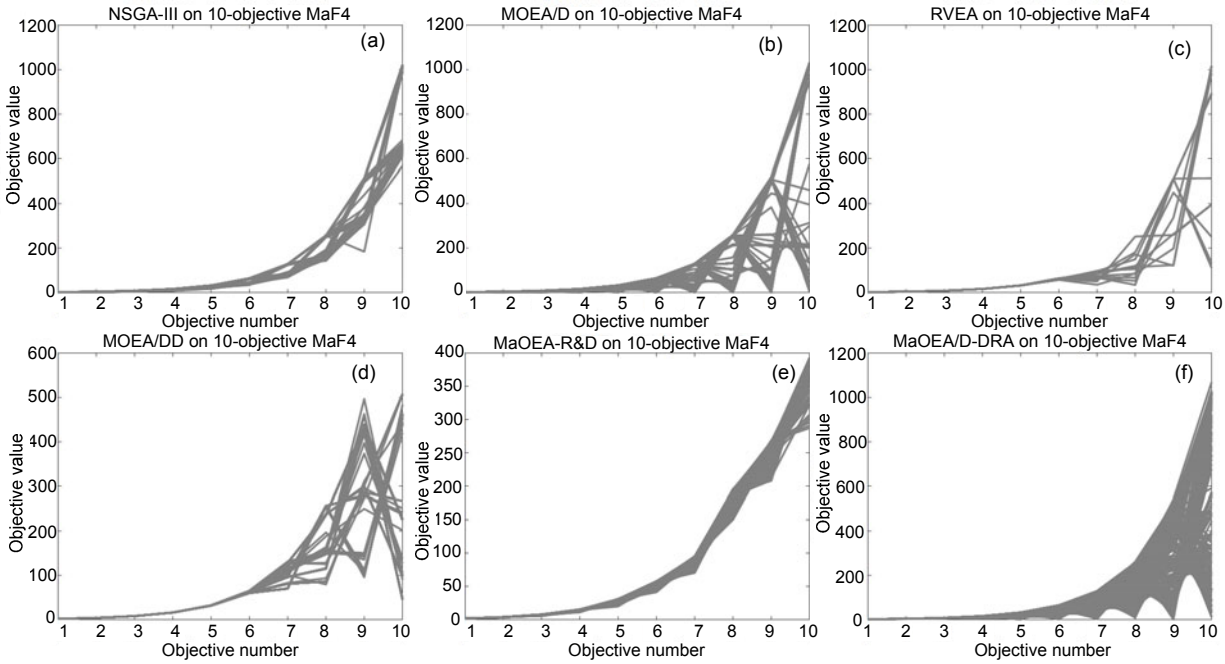

Fig. 5 Illustration of parallel coordinates of the non-dominated fronts on 10-objective MaF4, obtained by NSGA-III (a), MOEA/D (b), RVEA (c), MOEA/DD (d), MaOEA-R&D (e), and MaOEA/D-DRA (f) with the median IGD value in 30 runs

results obtained by MaOEA/D-DRA are only 8 and 10, respectively.

Overall, in terms of convergence speed and diversity, the solutions obtained by MaOEA/D-DRA are obviously better than those obtained by WS-MaOEA/D-DRA and PBI-MaOEA/D-DRA. Thus, the Tchebycheff approach is more suitable than the weighted-sum approach and penalty-based boundary intersection approach in the decomposition process of MaOEA/D-DRA for solving MaOPs with irregular PF.

4.5 Sensitivity to the number of previous learning generations *L* in MaOEA/D-DRA

In MaOEA/D-DRA, L is a parameter with a fixed value during the whole optimization process. In this subsection, three different values, i.e., 4, 8, and 12, are compared to study the sensitivity to L. For simplicity, MaOEA/D-DRA with three different values of 4, 8, and 12 are denoted as MaOEA/D-DRA-4, MaOEA/D-DRA-8, and MaOEA/D-DRA-12, respectively. Tables 7 and 8 present the IGD and HV values obtained by MaOEA/D-DRA-4, MaOEA/D-DRA-8, and MaOEA/D-DRA-12 over 30 independent runs in the three instances of DTLZ5, respectively, where the best results are highlighted. We observe that the results obtained by MaOEA/D-DRA with three different values of L are statistically similar to each other for the values of both IGD and HV. Therefore, we can conclude that, in the three test instances of DTLZ5, the performance of MaOEA/DRA is not sensitive to L.

5 Conclusions and future work

In this paper, we have proposed an MaOEA based on decomposition with DRA, termed MaOEA/D-DRA. In MaOEA/D-DRA, a new DRA method is designed to solve MaOPs with irregular PF (MaDRA), where MaOP is decomposed into a number of single optimization subproblems with the Tchebycheff approach, and each subproblem is associated with a solution in an evolutionary population. The computational resources are allocated to different subproblems dynamically based on the contribution of each subproblem to EA in the search process. Because the process involves solving MaOPs with more than three objectives, there is a large fraction of solutions in EA, so the method of SDE is used to update EA in the evolutionary process.

Table 5 IGD results of the Tchebycheff approach, weighted-sum approach, and penalty-based boundary intersection approach

Problem	M	IGD		
		WS-MaOEA/D-DRA	PBI-MaOEA/D-DRA	MaOEA/D-DRA
DTLZ5	5	1.4730e–1 (4.03e–2) –	**5.9574e–2 (8.42e–3) +**	6.7229e–2 (8.25e–3)
	8	2.6416e–1 (7.56e–2) –	**6.6762e–2 (1.44e–2) +**	1.3701e–1 (3.05e–2)
	10	2.5075e–1 (6.71e–2) –	**3.8660e–2 (1.13e–2) +**	1.1945e–1 (1.75e–2)
DTLZ6	5	2.0870e–1 (6.23e–2) –	**6.7547e–2 (8.98e–3) +**	9.0757e–2 (1.57e–2)
	8	2.9817e–1 (6.29e–2) –	**8.4188e–2 (2.75e–2) +**	1.4337e–1 (2.99e–2)
	10	2.9852e–1 (6.96e–2) –	**4.2073e–2 (1.80e–2) +**	1.2049e–1 (1.80e–2)
DTLZ7	5	4.1833e–1 (7.09e–2) –	6.9303e–1 (1.34e–1) –	**3.2191e–1 (4.32e–2)**
	8	**7.3380e–1 (2.69e–2) +**	1.6058e+0 (5.42e–1) –	7.6641e–1 (3.40e–2)
	10	1.0188e+0 (1.01e–1) ≈	2.3804e+0 (8.20e–1) –	**1.0055e+0 (1.09e–1)**
IDTLZ1	5	1.2796e–1 (1.50e–2) –	1.2824e–1 (4.02e–2) –	**7.0119e–2 (7.83e–4)**
	8	1.4305e–1 (9.39e–3) –	2.1996e–1 (2.84e–2) –	**1.1130e–1 (8.20e–4)**
	10	1.4513e–1 (8.66e–3) –	2.1279e–1 (1.79e–2) –	**1.2592e–1 (4.15e–3)**
IDTLZ2	5	2.2373e–1 (3.17e–3) –	2.6087e–1 (8.52e–3) –	**2.1889e–1 (4.87e–3)**
	8	**3.6103e–1 (3.31e–3) +**	4.8066e–1 (1.16e–2) –	3.6913e–1 (4.36e–3)
	10	**4.2558e–1 (2.95e–3) +**	5.6769e–1 (1.14e–2) –	4.2794e–1 (4.01e–3)
C1_DTLZ1	5	1.7025e–1 (1.07e–2) –	1.1737e–1 (3.35e–2) –	**9.3873e–2 (2.23e–2)**
	8	2.2379e–1 (3.21e–2) –	1.6215e–1 (1.21e–2) –	**1.4100e–1 (1.54e–2)**
	10	2.5588e–1 (3.69e–2) –	**1.6497e–1 (4.61e–3) ≈**	1.7481e–1 (2.44e–2)
WFG1	5	1.0712e+0 (3.28e–1) –	1.3687e+0 (2.30e–1) –	**5.9809e–1 (5.72e–2)**
	8	2.0784e+0 (4.53e–1) –	2.0766e+0 (3.51e–1) –	**1.0789e+0 (1.01e–1)**
	10	2.3381e+0 (4.10e–1) –	2.6750e+0 (1.92e–1) –	**1.5928e+0 (3.93e–1)**
WFG2	5	**1.2160e+0 (1.81e–1) ≈**	1.2997e+0 (2.24e–1) ≈	1.2829e+0 (1.46e–1)
	8	2.9951e+0 (4.55e–1) –	3.7285e+0 (4.64e–1) –	**2.5180e+0 (5.79e–1)**
	10	**3.8932e+0 (9.16e–1) ≈**	5.4506e+0 (5.42e–1) –	4.1714e+0 (1.03e+0)
WFG3	5	5.5539e–1 (6.15e–2) ≈	7.3240e–1 (8.66e–2) –	**5.3937e–1 (7.17e–2)**
	8	7.9676e–1 (7.60e–2) ≈	1.4592e+0 (1.37e–1) –	**7.4950e–1 (1.08e–1)**
	10	1.0151e+0 (1.16e–1) –	1.4872e+0 (6.68e–2) –	**8.6920e–1 (1.22e–1)**
MaF2	5	**1.1378e–1 (2.58e–3) +**	1.1871e–1 (2.30e–3) –	1.1677e–1 (2.05e–3)
	8	1.9600e–1 (1.36e–2) –	1.6742e–1 (3.10e–3) –	**1.6103e–1 (3.08e–3)**
	10	2.5239e–1 (1.73e–2) –	1.9441e–1 (3.65e–3) –	**1.7244e–1 (3.27e–3)**
MaF4	5	**2.5827e+0 (8.51e–2) +**	7.3668e+0 (8.36e–1) –	2.7955e+0 (1.65e–1)
	8	2.2172e+1 (1.02e+0) –	9.0864e+1 (5.37e+0) –	**2.1480e+1 (1.30e+0)**
	10	**7.4018e+1 (2.81e+0) +**	3.3082e+2 (3.19e+1) –	7.8675e+1 (3.34e+0)
MaF13	5	2.0682e–1 (8.18e–2) –	1.3517e–1 (1.97e–2) ≈	**1.3036e–1 (1.43e–2)**
	8	2.0557e–1 (3.24e–2) –	2.3362e–1 (4.86e–2) –	**1.3739e–1 (2.38e–2)**
	10	1.9602e–1 (3.21e–2) –	2.4963e–1 (4.73e–2) –	**1.3345e–1 (1.24e–2)**
+/–/≈		6/25/5	6/27/3	

M: number of objectives. The best result in each row is highlighted in bold

Table 6 HV results of the Tchebycheff approach, weighted-sum approach, and penalty-based boundary intersection approach

Problem	*M*	HV		
		WS-MaOEA/D-DRA	PBI-MaOEA/D-DRA	MaOEA/D-DRA
DTLZ5	5	7.3641e–3 (3.09e–4) –	**8.2108e–3 (1.55e–4) +**	8.0417e–3 (1.62e–4)
	8	1.6981e–5 (2.72e–7) –	**1.7808e–5 (3.82e–7) +**	1.7100e–5 (2.14e–7)
	10	5.6525e–8 (4.46e–10) ≈	**5.9029e–8 (7.52e–10) +**	5.6548e–8 (2.49e–10)
DTLZ6	5	7.0512e–3 (4.43e–4) –	**8.1071e–3 (2.14e–4) +**	7.7973e–3 (2.59e–4)
	8	1.6861e–5 (1.73e–7) –	**1.7724e–5 (5.33e–7) +**	1.7025e–5 (1.64e–7)
	10	5.6302e–8 (3.11e–10) –	**5.8881e–8 (1.20e–9) +**	5.6530e–8 (2.46e–10)
DTLZ7	5	1.8734e+0 (6.34e–2) –	3.3698e–1 (4.94e–1) –	**1.9345e+0 (7.84e–2)**
	8	**8.9156e–1 (4.06e–1) +**	2.3487e–3 (1.95e–3) –	4.7545e–1 (2.17e–1)
	10	**3.0498e–1 (2.94e–1) ≈**	2.8846e–4 (9.02e–4) –	2.2034e–1 (2.48e–1)
IDTLZ1	5	2.3811e–4 (3.28e–5) –	3.1171e–4 (1.18e–4) –	**4.6325e–4 (1.66e–5)**
	8	1.0143e–7 (3.39e–8) –	5.4653e–8 (3.34e–8) –	**1.6440e–7 (2.83e–8)**
	10	1.8649e–10 (5.72e–10) –	2.4008e–10 (1.00e–10) ≈	**7.8369e–10 (1.20e–9)**
IDTLZ2	5	1.8698e–1 (1.92e–3) +	**1.8780e–1 (1.87e–3) +**	1.7200e–1 (3.73e–3)
	8	5.6597e–3 (3.01e–4) +	**9.3738e–3 (3.00e–4) +**	5.1181e–3 (5.20e–4)
	10	1.3830e–4 (4.27e–5) –	**9.0345e–4 (1.35e–4) +**	2.3708e–4 (6.48e–5)
C1_DTLZ1	5	**4.2248e–2 (1.21e–3) ≈**	3.7470e–2 (4.37e–3) –	4.2046e–2 (3.42e–3)
	8	7.1830e–3 (2.14e–4) ≈	6.1300e–3 (4.39e–4) –	**7.3341e–3 (4.52e–4)**
	10	2.2014e–3 (7.98e–5) –	1.9972e–3 (1.22e–4) –	**2.3034e–3 (7.20e–5)**
WFG1	5	5.1404e+3 (5.77e+2) –	5.6433e+3 (2.56e+2) –	**6.0270e+3 (3.32e+0)**
	8	1.5214e+7 (2.24e+6) –	1.9110e+7 (1.56e+6) –	**2.0663e+7 (3.42e+5)**
	10	5.9908e+9 (1.15e+9) –	6.9765e+9 (1.07e+9) –	**7.6010e+9 (9.31e+8)**
WFG2	5	6.1179e+3 (1.39e+1) ≈	5.8929e+3 (4.74e+1) –	**6.1186e+3 (1.73e+1)**
	8	**2.2037e+7 (2.47e+4) ≈**	2.1197e+7 (3.59e+5) –	2.2024e+7 (3.92e+4)
	10	**9.6078e+9 (1.27e+7) ≈**	9.3157e+9 (8.20e+7) –	9.6043e+9 (1.41e+7)
WFG3	5	**2.3026e+0 (2.08e–1) +**	1.8482e–1 (2.99e–1) –	1.5790e+0 (2.87e–1)
	8	**1.8062e–2 (5.23e–3) +**	0.0000e+0 (0.00e+0) –	9.3936e–3 (6.33e–3)
	10	**3.6877e–5 (1.63e–5) +**	0.0000e+0 (0.00e+0) –	2.7001e–5 (1.67e–5)
MaF2	5	4.5010e–2 (8.23e–4) –	4.8698e–2 (3.25e–4) –	**5.0573e–2 (3.58e–4)**
	8	2.7864e–2 (4.93e–4) –	2.9891e–2 (3.19e–4) ≈	**2.9989e–2 (4.11e–4)**
	10	7.8478e–3 (1.23e–4) –	**8.2163e–3 (6.96e–5) +**	8.0826e–3 (8.82e–5)
MaF4	5	**3.4734e+3 (3.26e+2) +**	8.7420e+2 (4.47e+2) –	2.2155e+3 (5.79e+2)
	8	**4.6452e+7 (1.19e+7) +**	1.0656e+6 (3.45e+5) –	2.6243e+7 (1.52e+7)
	10	**1.5231e+11 (1.26e+11) ≈**	1.3022e+10 (3.52e+9) –	1.0429e+11 (4.90e+10)
MaF13	5	3.9381e–1 (6.95e–2) –	4.0153e–1 (1.45e–2) –	**4.2545e–1 (9.70e–3)**
	8	3.5848e–1 (8.66e–3) ≈	3.0013e–1 (3.75e–2) –	**3.5881e–1 (7.37e–3)**
	10	**3.6187e–1 (6.64e–3) ≈**	2.9947e–1 (4.88e–2) –	3.5999e–1 (4.94e–3)
+/–/≈		8/18/10	10/24/2	

M: number of objectives. The best result in each row is highlighted in bold

Table 7 IGD results of sensitivity to the number of previous learning generations *L* in MaOEA/D-DRA

Problem	*M*	IGD		
		MaOEA/D-DRA-4	MaOEA/D-DRA-12	MaOEA/D-DRA-8
DTLZ5	5	6.8958e–2 (8.75e–3) ≈	7.0693e–2 (1.02e–2) ≈	**6.8462e–2 (8.55e–3)**
	8	1.3346e–1 (2.13e–2) ≈	1.4333e–1 (3.16e–2) ≈	**1.2975e–1 (2.78e–2)**
	10	1.2062e–1 (1.81e–2) ≈	**1.1905e–1 (1.22e–2)** ≈	1.2222e–1 (1.96e–2)
+/–/≈		0/0/3	0/0/3	

M: number of objectives. The best result in each row is highlighted in bold

Table 8 HV results of sensitivity to the number of previous learning generations *L* in MaOEA/D-DRA

Problem	*M*	HV		
		MaOEA/D-DRA-4	MaOEA/D-DRA-12	MaOEA/D-DRA-8
DTLZ5	5	8.0344e–3 (1.62e–4) ≈	7.9592e–3 (1.71e–4) ≈	**8.0387e–3 (1.49e–4)**
	8	1.7096e–5 (1.66e–7) ≈	1.7041e–5 (1.99e–7) ≈	**1.7126e–5 (2.18e–7)**
	10	5.6536e–8 (3.52e–10) ≈	**5.6593e–8 (2.53e–10)** ≈	5.6561e–8 (2.69e–10)
+/–/≈		0/0/3	0/0/3	

M: number of objectives. The best result in each row is highlighted in bold

To assess the performance of MaOEA/D-DRA, empirical comparisons have been conducted by comparing MaOEA/D-DRA with five state-of-the-art MaOEAs, namely NSGA-III, MOEA/D, RVEA, MOEA/DD, and MaOEA-R&D, using widely used MaOP test problems with irregular PF. Two different metrics, namely IGD and HV, have been calculated and compared among all the algorithms. The Wilcoxon rank-sum tests were conducted to show statistically significant differences between all the algorithms. Based on the numerical results, MaOEA/D-DRA is superior to the five algorithms compared generally.

In addition, the effectiveness of the Tchebycheff approach is proved by comparison with the weighted-sum approach and penalty-based boundary intersection approach in the same algorithmic framework. In terms of IGD and HV, the results of Wilcoxon rank-sum tests showed that the Tchebycheff approach outperforms the two other decomposition methods as a whole.

MaOEA/D-DRA has shown competitive performance in the studied MaOPs with irregular PF. However, MaOEA/D-DRA does not perform well on the constrained problem C1_DTLZ1. Therefore, investigations into application of MaOEA/D-DRA for the constrained MaOPs is one future line of work. Furthermore, real-life applications of MaOEA/D-DRA need to be investigated.

Contributors

Ming-gang DONG guided the research. Bao LIU designed the research and drafted the manuscript. Chao JING helped organize the manuscript. Ming-gang DONG and Chao JING revised and finalized the paper.

Compliance with ethics guidelines

Ming-gang DONG, Bao LIU, and Chao JING declare that they have no conflict of interest.

References

Asafuddoula M, Ray T, Sarker R, 2015. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans Evol Comput*, 19(3):445-460. https://doi.org/10.1109/TEVC.2014.2339823

Cai XY, Li YX, Fan Z, et al., 2015. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Trans Evol Comput*, 19(4):508-523. https://doi.org/10.1109/TEVC.2014.2350995

Cai XY, Yang ZX, Fan Z, et al., 2017. Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. *IEEE Trans Cybern*, 47(9):2824-2837. https://doi.org/10.1109/TCYB.2016.2586191

Cai XY, Mei ZW, Fan Z, et al., 2018. A constrained decomposition approach with grids for evolutionary multiobjective optimization. *IEEE Trans Evol Comput*, 22(4):564-577. https://doi.org/10.1109/TEVC.2017.2744674

Chand S, Wagner M, 2015. Evolutionary many-objective optimization: a quick-start guide. *Surv Oper Res Manag Sci*, 20(2):35-42. https://doi.org/10.1016/j.sorms.2015.08.001

Cheng R, Jin YC, Olhofer M, et al., 2016. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput*, 20(5):773-791. https://doi.org/10.1109/TEVC.2016.2519378

Cheng R, Li MQ, Tian Y, et al., 2018. Benchmark Functions for the CEC'2018 Competition on Many-Objective Optimization. University of Birmingham, United Kingdom.

Coello CAC, 2006. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Comput Intell Mag*, 1(1):28-36. https://doi.org/10.1109/MCI.2006.1597059

Coello CAC, Lechuga MS, 2002. MOPSO: a proposal for multiple objective particle swarm optimization. Proc Congress on Evolutionary Computation, p.1051-1056. https://doi.org/10.1109/CEC.2002.1004388

Corne DW, Jerram NR, Knowles JD, et al., 2001. PESA-II: region-based selection in evolutionary multiobjective optimization. Proc 3rd Annual Conf on Genetic and Evolutionary Computation, p.283-290.

Das I, Dennis JE, 1998. Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optim*, 8(3): 631-657. https://doi.org/10.1137/S1052623496307510

Deb K, Goyal M, 1996. A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform*, 26(4):30-45.

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. https://doi.org/10.1109/TEVC.2013.2281535

Deb K, Pratap A, Agarwal S, et al., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*, 6(2):182-197. https://doi.org/10.1109/4235.996017

Deb K, Thiele L, Laumanns M, et al., 2005. Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (Eds.), Evolutionary Multiobjective Optimization. Springer, London, p.105-145. https://doi.org/10.1007/1-84628-137-7_6

Elarbi M, Bechikh S, Gupta A, et al., 2018. A new decomposition-based NSGA-II for many-objective optimization. *IEEE Trans Syst Man Cybern Syst*, 48(7):1191-1210. https://doi.org/10.1109/TSMC.2017.2654301

He C, Tian Y, Jin YC, et al., 2017. A radial space division based evolutionary algorithm for many-objective optimization. *Appl Soft Comput*, 61:603-621. https://doi.org/10.1016/j.asoc.2017.08.024

He ZN, Yen GG, 2016. Many-objective evolutionary algorithm: objective space reduction and diversity improvement. *IEEE Trans Evol Comput*, 20(1):145-160. https://doi.org/10.1109/tevc.2015.2433266

Huband S, Hingston P, Barone L, et al., 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput*, 10(5):477-506. https://doi.org/10.1109/tevc.2005.861417

Jain H, Deb K, 2014. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Trans Evol Comput*, 18(4):602-622. https://doi.org/10.1109/TEVC.2013.2281534

Kukkonen S, Lampinen J, 2005. GDE3: the third evolution step of generalized differential evolution. IEEE Congress on Evolutionary Computation, p.443-450. https://doi.org/10.1109/CEC.2005.1554717

Li BD, Li JL, Tang K, et al., 2015. Many-objective evolutionary algorithms: a survey. *ACM Comput Surv*, 48(1):13. https://doi.org/10.1145/2792984

Li H, Zhang QF, 2009. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput*, 13(2):284-302. https://doi.org/10.1109/TEVC.2008.925798

Li K, Deb K, Zhang QF, et al., 2015. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput*, 19(5):694-716. https://doi.org/10.1109/TEVC.2014.2373386

Li MQ, Yang SX, Liu XH, 2014. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans Evol Comput*, 18(3):348-365. https://doi.org/10.1109/tevc.2013.2262178

Li MQ, Yang SX, Liu XH, 2015. Bi-goal evolution for many-objective optimization problems. *Artif Intell*, 228:45-65. https://doi.org/10.1016/j.artint.2015.06.007

Liu YP, Gong DW, Sun XY, et al., 2017. Many-objective evolutionary optimization based on reference points. *Appl Soft Comput*, 50:344-355. https://doi.org/10.1016/j.asoc.2016.11.009

Purshouse RC, Fleming PJ, 2007. On the evolutionary optimization of many conflicting objectives. *IEEE Trans Evol Comput*, 11(6):770-784. https://doi.org/10.1109/TEVC.2007.910138

Qi YT, Ma XL, Liu F, et al., 2014. MOEA/D with adaptive weight adjustment. *Evol Comput*, 22(2):231-264. https://doi.org/10.1162/EVCO_a_00109

Ruan WY, Duan HB, 2020. Multi-UAV obstacle avoidance control via multi-objective social learning pigeon-inspired optimization. *Front Inform Technol Electron Eng*, 21(5):740-748. https://doi.org/10.1631/FITEE.2000066

Santiago A, Huacuja HJF, Dorronsoro B, et al., 2014. A survey of decomposition methods for multi-objective optimization. In: Castillo O, Melin P, Pedrycz W (Eds.), Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Springer, Cham, p.453-465. https://doi.org/10.1007/978-3-319-05170-3_31

Seada H, Deb K, 2016. A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Trans Evol Comput*, 20(3):358-369. https://doi.org/10.1109/TEVC.2015.2459718

Seada H, Abouhawwash M, Deb K, 2019. Multiphase balance of diversity and convergence in multiobjective optimization. *IEEE Trans Evol Comput*, 23(3):503-513. https://doi.org/10.1109/TEVC.2018.2871362

Tian Y, Cheng R, Zhang XY, et al., 2017. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [Educational Forum]. *IEEE Comput Intell Mag*, 12(4):73-87. https://doi.org/10.1109/MCI.2017.2742868

Tian Y, Cheng R, Zhang XY, et al., 2018. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans Evol Comput*, 22(4):609-622. https://doi.org/10.1109/TEVC.2017.2749619

Trivedi A, Srinivasan D, Sanyal K, et al., 2017. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Trans Evol Comput*, 21(3):440-462. https://doi.org/10.1109/TEVC.2016.2608507

Wang GP, Jiang HW, 2007. Fuzzy-dominance and its application in evolutionary many objective optimization. Int Conf on Computational Intelligence and Security Workshops, p.195-198. https://doi.org/10.1109/CISW.2007.4425478

Wang TC, Ting CK, 2018. Fitness inheritance assisted MOEA/D-CMAES for complex multi-objective optimization problems. IEEE Congress on Evolutionary Computation, p.1-8. https://doi.org/10.1109/CEC.2018.8477898

Wen XY, Chen WN, Lin Y, et al., 2017. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Trans Evol Comput*, 21(3):363-377. https://doi.org/10.1109/TEVC.2016.2605501

Yang SX, Li MQ, Liu XH, et al., 2013. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput*, 17(5):721-736. https://doi.org/10.1109/TEVC.2012.2227145

Zeng YJ, Sun YG, 2014. Comparison of multiobjective particle swarm optimization and evolutionary algorithms for optimal reactive power dispatch problem. IEEE Congress on Evolutionary Computation, p.258-265. https://doi.org/10.1109/CEC.2014.6900260

Zhang HP, Hui Q, 2019. Many objective cooperative bat searching algorithm. *Appl Soft Comput*, 77:412-437. https://doi.org/10.1016/j.asoc.2019.01.033

Zhang L, Zhang JY, Li T, et al., 2017. Multi-objective aerodynamic optimization design of high-speed train head shape. *J Zhejiang Univ-Sci A (Appl Phys & Eng)*, 18(11): 841-854. https://doi.org/10.1631/jzus.A1600764

Zhang QF, Li H, 2007. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput*, 11(6):712-731. https://doi.org/10.1109/TEVC.2007.892759

Zhang QF, Liu WD, Tsang E, et al., 2010. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans Evol Comput*, 14(3):456-474. https://doi.org/10.1109/TEVC.2009.2033671

Zhang XY, Tian Y, Jin YC, 2015. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput*, 19(6):761-776. https://doi.org/10.1109/TEVC.2014.2378512

Zhou AM, Jin YC, Zhang QF, et al., 2006. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. IEEE Int Conf on Evolutionary Computation, p.892-899. https://doi.org/10.1109/CEC.2006.1688406

Zhu QL, Zhang QF, Lin QZ, et al., 2019. MOEA/D with two types of weight vectors for handling constraints. IEEE Congress on Evolutionary Computation, p.1359-1365. https://doi.org/10.1109/CEC.2019.8790336

Zitzler E, Künzli S, 2004. Indicator-based selection in multi-objective search. Proc 8th Int Conf on Parallel Problem Solving from Nature, p.832-842. https://doi.org/10.1007/978-3-540-30217-9_84

Zitzler E, Laumanns M, Thiele L, 2001. SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report, Vol. 103. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK). https://doi.org/10.3929/ethz-a-004284029

Zou XF, Chen Y, Liu MZ, et al., 2008. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Trans Syst Man Cybern*, 38(5):1402-1412. https://doi.org/10.1109/TSMCB.2008.926329