University of Birmingham Research at Birmingham

Solving many-objective optimization problems by a Pareto-based evolutionary algorithm with preprocessing and a penalty mechanism

Liu, Yuan; Zhu, N.; Li, Miqing

DOI:
10.1109/TCYB.2020.2988896

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Liu, Y, Zhu, N & Li, M 2021, 'Solving many-objective optimization problems by a Pareto-based evolutionary algorithm with preprocessing and a penalty mechanism', *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5585-5594. https://doi.org/10.1109/TCYB.2020.2988896

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
Y. Liu, N. Zhu and M. Li, "Solving Many-Objective Optimization Problems by a Pareto-Based Evolutionary Algorithm With Preprocessing and a Penalty Mechanism," in IEEE Transactions on Cybernetics, vol. 51, no. 11, pp. 5585-5594, Nov. 2021, doi: 10.1109/TCYB.2020.2988896.

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of ‘fair dealing’ under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 20. Apr. 2024

Solving Many-Objective Optimization Problems by a Pareto-based Evolutionary Algorithm with Preprocessing and a Penalty Mechanism

Yuan Liu, Ningbo Zhu and Miqing Li

***Abstract*—It is known that the Pareto-based approach is not well suited for optimization problems with a large number of objectives, even though it is a class of mainstream methods in multi-objective optimization. Typically, a Pareto-based algorithm comprises two parts: a Pareto dominance-based criterion and a diversity estimator. The former guides the selection toward the optimal front, while the latter promotes the diversity of the population. However, the Pareto dominance-based criterion becomes ineffective in solving optimization problems with many objectives (e.g., more than 3), and thus, the diversity estimator will determine the performance of the algorithm. Unfortunately, the diversity estimator usually has a strong bias toward dominance resistance solutions (DRSs), thereby failing to push the population forward. DRSs are solutions that are far away from the Pareto optimal front but cannot be easily dominated. In this paper, we propose a new Pareto-based algorithm to resolve the above issue. First, to eliminate the DRSs, we design an interquartile range method to preprocess the solution set. Second, to balance convergence and diversity, we present a penalty mechanism of alternating operations between selection and penalty. The proposed algorithm is compared with five state-of-the-art algorithms on a number of well-known benchmarks with 3-15 objectives. The experimental results show that the proposed algorithm can perform well on most of the test functions and generally outperforms its competitors.**

***Keywords*—Many-objective optimization, evolutionary algorithm, dominance resistance solutions.**

I. Introduction

RECENTLY, many-objective problems (MaOPs) with at least four conflicting objectives have drawn increasing attention in evolutionary computation. MaOPs often appear in real-world applications, such as directed acyclic graph scheduling [1], workflow applications [2], and task secheduling [3], as a result, algorithms in the evolutionary computation field, called many-objective evolutionary algorithms (MaOEAs), have been rapidly developed. In general, such an algorithm searches for a set of well-converged and well-distributed trade-off solutions instead of a single optimal solution in single-objective optimization.

Conventional evolutionary algorithms, which use Pareto dominance as the primary selection criterion, may perform very poorly in solving MaOPs [4], [5]. A commonly accepted notion is that due to the “curse of dimensionality”, Pareto-based algorithms encounter two significant challenges. The first is the dominance resistant phenomenon that leads to incomparability of solutions [6]. The resulting solutions, called dominance resistance solutions (DRSs), survive over numerous generations (or even exist in the final population). DRSs are solutions that are far away from the Pareto-optimal front but cannot be easily dominated. The second is the difficulty of maintaining a good diversity in a high-dimensional space [7], [8]. Diversity is subdivided into two aspects: uniformity and spread. Uniformity quantifies the distance between neighbouring points in solutions; however, the traditional distance metrics (e.g., Euclidean distance-based estimators) in a very high-dimensional objective space become less reliable. The spread refers to the coverage of solutions in the objective space; however, a large search space and the limited number of solutions certainly conflict with each other.

To overcome the above problems, researchers have focused on the development of new algorithms and techniques [8], [9] that can be roughly divided into the following four categories.

The first approach involves dominance relaxation. If Pareto dominance is unsuitable for many-objective optimization, a straightforward idea is to modify Pareto dominance by enlarging the dominance areas of a solution to increase the pressure on the selection toward the Pareto front. Several related approaches have been proposed, including ϵ-dominance [10], [11], controlling dominance area of a solution [12], and Fuzzy Pareto dominance [13], [14], and others [15]–[17]. Unfortunately, all of these approaches inevitably entail difficulties in determining the extent of relaxation for different problems, which has led to the emergence of dynamic tuning methods [18], [19].

The second approach is diversity-based. In many-objective optimization, the diversity selection mechanism plays a vital role when Pareto dominance cannot provide sufficient pressure on the selection toward the Pareto-optimal front. In view of this, another way of adapting Pareto-based algorithms

This work was supported in part by the National Natural Science Foundation of China under Grant 61572177, the National Outstanding Youth Science Program of National Natural Science Foundation of China under Grant 61625202 and the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China under Grant 61661146006, the Postgraduate Scientific Research Inovation Project of Hunan under Grant CX20190309. (*Corresponding author: Ningbo Zhu*)

Yuan Liu and Ningbo Zhu are with the College of Computer Science and Electronic Engineering, Hunan University, Hunan, China. (email:{liu3yuan, quietwave}@hnu.edu.cn).

Miqing Li is with the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (email:limitsing@gmail.com).

for MaOPs is to modify their diversity maintenance strategy [20]. Specifically, in [21], the crowding distance of boundary solutions of NSGA-II is assigned a zero value to improve the proximity of the algorithm; in [22], a diversity management mechanism can be switched off when the population is excessively diverse, and in [23], a shift-based density estimation (SDE) mechanism is used to shift the position of solutions in their density estimation to reflect their convergence.

The indicator-based approach is the third type. Indicators are commonly used to evaluate the performance of algorithms in the area [24]. This approach uses a performance indicator to calculate the fitness of solutions and guide the search process. Some well-known MOEAs based on the indicator-based approach are IBEA [25], SMS-MOEA [26], and HypE [27]. However, the calculation time of hypervolume-based algorithms (such as SMS-MOEA and HypE) is usually longer than that of other algorithms.

The last category is the decomposition-based approach. In this approach, a scalarizing function is used to decompose a MaOP into a set of single-objective subproblems. A representative example is the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [28]. MOEA/D decomposes a MaOP into a number of subproblems and uses a set of evenly distributed weights to maintain the diversity of the population. Many algorithms based on decomposition, such as PAEA [29], MOEA/D-LWS [30], MOEA/DD [31], have recently been proposed to deal with many-objective problems. Additionally, adaptive weight setting has become a research hotspot in this category in recent years, and it used in methods such as MOEA/HD [32], A-NSGA-III [33] and RVEA [34].

Although these studies have enhanced the search ability of MaOEAs and provided a variety of approaches to MaOPs, the field of many-objective optimization is far from maturity. Notably, relaxation factor tuning is the most challenging obstacle in the use of the first approach. Due to the uncertain environment of a given problem, it is difficult for the diversity-based approach to balance the convergence and diversity of solutions in the many-objective space. As described above, the indicator-based approach requires time-consuming computations to obtain the indicator values of all solutions. For the decomposition-based approach, the challenge is to specify a set of appropriate reference points that are consistent with the Pareto front's shape of a given problem, although some results have been derived in this regard.

In this paper, we focus on diversity-based approaches. When solving MaOPs, these approaches are intended to improve performance by reducing the adverse effects of diversity maintenance. However, these approaches' performance aspects of convergence and diversity are difficult to reconcile under an uncertain environment, and performance remains far from meeting the requirements of MaOPs. On the one hand, premature convergence is considered one of their recurrent drawbacks [35]. It is generally due to excessive selection pressure and the loss of diversity within the population. In this case, it is challenging for the algorithm to escape the local optimum represented by the population [36]. On the other hand, many schemes for avoiding premature convergence have been devised in an effort to increase the diversity of population [37]. However, if the population is too diverse, a large number of DRSs will exist in the population when MaOPs are being solved, resulting in a slow convergence and eventually producing poor-quality solutions. Naturally, existing studies have explored various methods for detecting DRSs. For example, Ikeda *et al.* [38] proposed a relaxed form of the dominance method, called α-dominance, for dominating the DRSs. G. Yu *et al.* [39] introduced a novel boundary elimination selection based on binary search trying to avoid the impact of DRSs during optimization. Bhattacharjee *et al.* [40] proposed a Six-Sigma-based method for removing the influence of DRSs on nadir point calculation. Finally, W. Hu *et al.* [41] proposed using the mean absolute deviation method to detect DRSs.

To alleviate the above challenges, this paper proposes a novel Pareto-based evolutionary algorithm (named PMEA) using a penalty mechanism to optimize the preprocessed population. The basic idea of PMEA is that it alternately selects and penalizes solutions to ensure population quality. We first select the best solution from preprocessed solutions based on a convergence metric and subsequently penalize its neighbouring solutions according to a distribution metric. Note that the penalty operation means deprioritizing solutions' entries rather than deleting solutions. Additionally, we introduce an interquartile range method (IQR) as a preprocessing operation to eliminate the DRSs.

The main contributions of this paper can be summarized as follows.

- An effective preprocessing mechanism is proposed. In this paper, we explore the impact of DRSs on algorithms used for optimization MaOPs and emphasize the importance of detecting and removing DRSs. Furthermore, we introduce a new DRS detection method, and verify the effectiveness of the method through experiments.
- A novel diversity maintenance method is proposed. In this method, the convergence and diversity maintenance operations are alternated. Once the solution with the best convergence has been selected, its diversity will be considered. An important feature of this method is that it weakens the sensitivity to the distribution threshold during the diversity maintenance process.
- We propose a new Pareto-based algorithm that integrates the above two strategies. Extensive experiments are performed to assess the performance of our algorithm in solving MaOPs by comparing it with several state-of-the-art MOEAs for MaOPs on three suites of widely used test problems.

The rest of this paper is organized as follows. Section II presents the related studies and outlines the motivation of this research. Section III describes the proposed algorithm in detail. In Section IV, extensive experiments are performed to compare the proposed algorithm with five state-of-the-art algorithms. Finally, we draw conclusions in Section V.

II. Related studies and Motivation

To introduce the background knowledge of this paper, the impact of DRSs and several diversity-based mechanisms are

TABLE I. SOLUTION VALUES IN THE BI-OBJECTIVE SPACE.

	A	**B**	**C**	**D**	**E**	**F**
Objective space	(100, 0)	(0.8, 0.2)	(0.6, 0.4)	(0.4, 0.6)	(0.2, 0.8)	(0, 1)
Normalized objective space	(1, 0)	(0.008, 0.2)	(0.006, 0.4)	(0.004, 0.6)	(0.002, 0.8)	(0, 1)

briefly reviewed. The inspirations of the proposed algorithm are also presented at the end of each related topic.

A. Impact of DRSs

In general, Pareto-based algorithms consider non-dominated solutions as proper candidate solutions to be kept. If the population size limit is exceeded, some of the more crowded solutions will be removed based on another criterion, e.g., density estimation. However, if there are solutions that are hardly dominated but significantly worse than others and that are spread among the population, then they cannot be cleaned by Pareto-based strategies [38], [42]. This phenomenon can be regarded as dominance resistance, and these solutions are called dominance resistance solutions (DRSs).

Numerically, the term DRSs refers to non-dominated solutions with a poor value in at least one of the objectives but with nearly optimal values in the other objectives; such solutions are usually located in the boundary regions. Since DRSs are boundary solutions, most of them will have the best density value, resulting in these solutions being always included in the new population [42]. Additionally, it is generally recognized that the proportion of non-dominated solutions in a population increases exponentially with the number of objectives [43]. As a result, the number of DRSs may also increase with the number of objectives, causing a more intense dominance resistance [44]. Thus the existence of these solutions will hinder the population's progress toward the Pareto-optimal front since their values in several objectives are relatively poor.

DRSs can seriously inflate the nadir in specific axial directions, resulting in uneven scaling of each objective. Many MaOEAs [45]–[47] considered in the literature typically use a normalization operation to map the objective values of the non-dominated set to the same interval (e.g., $[0, 1]$) using their ranges before the environmental selection to facilitate solving the MaOPs with various objectives scaled to different ranges. Most often, the min-max normalization that performs a linear transformation on the original solutions is adopted in the evolutionary computation. The minimum and maximum values of each objective are the corresponding elements in the ideal point and the nadir point, respectively. However, the presence of DRSs may result in the effective range of each normalized objective not necessarily being in the range $[0, 1]$. For example, assume that there is a set of non-dominated solutions in the bi-objective space, which contains a DRS with a poor value in objective f_1, as shown in Table 2. After normalization, the effective ranges of objectives f_1 and f_2 are scaled to the region $[0.01, 0]$ and the range $[1, 0]$, respectively. Additionally, the DRS may be more easily preserved in the normalized space. As shown in the table, solutions $\mathbf{A}$ and $\mathbf{F}$ have the same convergence in the normalized space, but the distribution of the former is significantly better than that of the latter.

Moreover, if there is a large number of DRSs in the population, the offspring they generate through the recombination operator are also most likely to be DRSs. In many-objective problems, the limited size of the population causes solutions to be far apart from each other [47]. In such a population, two distant parent solutions are likely to produce offspring solutions that are also distant from parents [48]. In this scenario, the effect of the recombination operator questionable. Therefore, most MaOEAs [34], [47] typically use simulated binary crossover (SBX) with a large distribution index, leading to a high probability of sampling an offspring close to its parents. However, this approach of emphasizing near-parent solutions may further increase the number of DRSs in the new population if there are some DRSs in the mating pool.

Given all that, it is undeniable that a reasonable DRS detection mechanism will be invaluable for the many-objective optimization process. The detection and elimination of DRSs is not easy, and further research is required to address this issue. In this paper, we introduce an interquartile range method that eliminates all non-dominated solutions with convergence metrics that are are in the upper quartile.

B. Niche-based diversity maintenance mechanisms

For Pareto-based algorithms, niche-based techniques–a common method of solving MaOPs–can achieve a performance trade-off by controlling the distribution of well-converged solutions [45], [46], [49], [50]. In this section, we review some classic niche-based maintenance mechanisms that inspired the proposed penalty mechanism.

A knee point driven MOEA proposed in [49] prefers knee points of the non-dominated fronts in selection. Specifically, KnEA uses the solution farthest from a hyperplane in a local region as the knee point, and the hyperplane is defined by M extreme points in the non-dominated solutions. Whenever a knee point is confirmed, its neighbours will be penalized. KnEA uses a hyperbox to identify the neighbours of a knee point, and the size of the hyperbox can be adaptively changed during the evolution.

The 1by1EA [46] method was the first to introduce the one-by-one selection strategy. The main idea is that during the environmental selection, well-converged solutions are selected one-by-one based on a computationally efficient convergence metric. Once the best-converged solution has been selected, its neighbours are penalized using a niche technique to guarantee the diversity of the population. In 1by1EA, the neighbours of a solution are determined by cosine similarity, and the distribution threshold is adaptively updated similarly to the approach used in KnEA.

VaEA [45] emphasizes a new optimizer based on the search directions of the evolving population itself. It first uses the maximum-vector-angle-first principle to select the better-density solutions from the non-dominated set to build the search directions and then uses the worse-elimination principle to find the better-convergence solutions in those search directions. In VaEA, the threshold of each search direction is set to a fixed value that is only related to the size of the population.

Intuitively, the above algorithms all use a niche preservation technique to keep a balance between convergence and diversity.

An effective way to determine the correct threshold becomes the key to achieving good diversity performance for such algorithms, however, this task is quite difficult. If the threshold is too large, most of the solutions will be penalized so that the selection operation cannot be completed in one round, and the end of a round will occur when the candidate set is empty. Multiple rounds of selection will result in numerous adjacent solutions being selected into the next population. Conversely, if the threshold is too small, the next population will be filled before the end of the first round of selection. As a result, it will be difficult for the population to cover the entire Pareto-optimal front. As analysed in [50], the most desirable state is to complete the selection of the next population if and only if the candidate set is empty for the first time. However, it is difficult for the existing adaptive threshold mechanism to attain such a state. In this paper, we propose a new mechanism to achieve this goal by controlling the number of penalized solutions, where the number is equal to the difference between the size of the candidate set and the number of selected solutions in each generation.

III. Proposed Method

The proposed method consists of preprocessing and a penalty mechanism. Preprocessing eliminates DRSs in the population, and the penalty mechanism selects a set of well-converged and well-distributed trade-off solutions from the preprocessed population.

A. Pre-processing

The pre-processing is an essential step in the data mining process, and the removal of outliers is one of its purposes [51]. In fact, numerous outliers (or DRSs) occur in the population during optimization many-objective problems. As discussed in Section II-A, such DRSs can have an adverse effect on the performance of the population.

In this paper, we introduce an interquartile range (IQR) method to detect and eliminate DRSs. IQR, being a descriptive statistic, can be applied to any data. Given an ordered set of one-dimensional data $\mathbf{X}$, we first determine low quartile Q_1 and upper quartile Q_3, and subsequently calculate the upper and lower boundaries of the data. Formally,

$$\begin{cases} UB = Q_3 + r\,(Q_3 - Q_1) \\ LB = Q_1 - r\,(Q_3 - Q_1) \end{cases}, \tag{1}$$

where UB and LB represent the upper and lower boundaries of the valid dataset, respectively. The sensitivity of r on the performance of our algorithm is explored in Section I of the supplementary file and it was eventually set to 1.5. In addition, Q_1 and Q_3 are defined as

$$\begin{cases} Q_1 = \mathbf{X}_{\lfloor(\frac{|\mathbf{X}|}{4})\rfloor}\left(1 - \left(\frac{|\mathbf{X}|}{4} - \left\lfloor\left(\frac{|\mathbf{X}|}{4}\right)\right\rfloor\right)\right) + \\ \qquad \mathbf{X}_{\lfloor(\frac{|\mathbf{X}|}{4})\rfloor+1}\left(\frac{|\mathbf{X}|}{4} - \left\lfloor\left(\frac{|\mathbf{X}|}{4}\right)\right\rfloor\right) \\ Q_3 = \mathbf{X}_{\lfloor(\frac{3|\mathbf{X}|}{4})\rfloor}\left(1 - \left(\frac{3|\mathbf{X}|}{4} - \left\lfloor\left(\frac{3|\mathbf{X}|}{4}\right)\right\rfloor\right)\right) + \\ \qquad \mathbf{X}_{\lfloor(\frac{3|\mathbf{X}|}{4})\rfloor+1}\left(\frac{3|\mathbf{X}|}{4} - \left\lfloor\left(\frac{3|\mathbf{X}|}{4}\right)\right\rfloor\right) \end{cases}, \tag{2}$$

Algorithm 1: Pre-processing

```
   Input: Solution Set: $\mathbf{F}$, Population Size: $n$
   Output: Solution Set: $\overline{\mathbf{F}}$
1  Compute the convergence metric of solutions $\mathbf{F}$: $\mathbf{CM}$;
2  Sort $\mathbf{CM}$;
3  Compute $Q_1$ and its index $i$;
4  Compute $Q_3$ and its index $j$;
5  if $n > j$ then
6      $i = size(\mathbf{CM}) - n$ and $Q_1 = \mathbf{CM}(i)$;
7      $j = n$ and $Q_3 = \mathbf{CM}(j)$;
8  end
9  Compute the upper boundary (UB) of the ordered set
     $\mathbf{CM}$;
10 Eliminate the solutions whose convergence values are
     greater than the UB;
```

where $\lfloor\cdot\rfloor$ indicates rounding down and $|\cdot|$ represents the size of the set.

Outliers are points that are on either end of the dataset. For example, consider dataset $\mathbf{X} = \{5, 40, 42, \cdots, 58, 60, 95\}$ in Fig. 1. According to (1), the blue points $\{5, 95\}$ are outside the boundary, and are thus regarded as outliers.

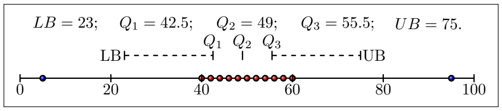
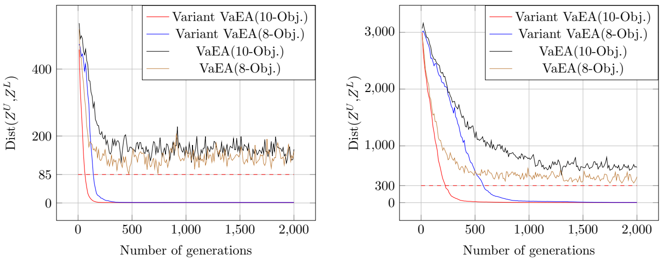


Fig. 1. An example of the interquartile method, where LB and UB represent the lower and upper boundaries, respectively, and Q_2 and Q_3 represent the lower and upper quartiles, respectively.

In evolutionary computation, as outliers are more significant than the rational solutions in terms of the convergence metric, we only need to eliminate the solutions with convergence metrics that are greater than the upper boundary. The pseudocode of the preprocessing operation is shown in Algorithm 1, which assumes that there is a set of non-dominated solutions $\mathbf{F}$ that outnumber the population size n. First, we calculate the convergence metric for each solution in the non-dominated set and keep it in $\mathbf{CM}$. Afterwards, we need to sort the values of $\mathbf{CM}$ in the ascending order. Next, the upper and lower quartiles (Q_3 and Q_1) of the ordered set $\mathbf{CM}$ are calculated by Eq. (2), and their index positions are recorded as i and j, respectively. To prevent the number of candidate solutions after preprocessing from being less than the population size n, we compare the size of Q_3's index j with the size n of the population n. If $j < n$, we adjust the upper and lower quartiles of $\mathbf{CM}$ in steps 6 and 7 of the algorithm. After the upper and lower quartiles have been determined, the upper boundary ($\mathbf{UB}$) is calculated according to Eq. (1). Finally, we eliminate all solutions with convergence metrics that are on the upper boundary.

Fig. 2 shows a comparison trajectories of the distance between the nadir point (Z^U) and the ideal point (Z^L) of the original VaEA [45] method and the variant with IQR



(a) DTLZ1

(b) DTLZ3

Fig. 2. Distance between the nadir point (Z^U) and the idea point (Z^L) in each generation of the runs on the (a) 10- and 8-objective DTLZ1 problems and (b) 10- and 8-objective DTLZ3 problems, respectively

on the DTLZ1 and DTLZ3 problems [52]. VaEA is selected because it is a Pareto-based algorithm, and its performance is heavily impacted by DRSs. Here Z^U is composed of the maximum value of each objective for all solutions, and Z^L is set to the coordinate origin. As shown in Fig. 2, the values of Dist(Z^U, Z^L) of the two algorithms decrease gradually and eventually fluctuate around a horizontal line, but Z^U of VaEA is significantly more bloated and turbulent than that of the variant version. Additionally, this figure shows that the convergence of the variant version is much better than that of the original VaEA.

B. Penalty Mechanism

After the candidate solution set has been preprocessed, our penalty mechanism takes into account the convergence and diversity of the population to select a set of superior solutions for the next generation. First, we need to obtain the convergence and distribution metrics for each candidate solution and keep them in sets $\mathbf{CM}$ and $\mathbf{DM}$, respectively. Next, the best solution is selected from the candidate set for the next generation according to the obtained convergence metrics $\mathbf{CM}$. Once a solution has been, we need to distinguish its neighbours according to the distribution metrics $\mathbf{DM}$ and penalize them into the penalty set $\mathbf{E}$. If the size of $\mathbf{E}$ exceeds its upper limit determined by the difference between the size of the initial candidate set (s) and the size of the required population (n), we should choose some of the well-distributed solutions from the penalty set as candidate solutions by comparing the similarity between the solutions in the penalty set and the selected solutions. Here, the top-(s-n) solutions closest to the selected solutions are regarded as neighours of the latter, and the other solutions in the penalty set are moved into the candidate set as well-distributed solutions. Finally, during selection, the solutions in the penalty set are always closest to the selected solutions, and their count is less than or equal to the maximum capacity of the penalty set. The procedure is presented in Algorithm 2. To describe the penalty mechanism more clearly, we give a simple example in section II of the supplementary file.

As described above, the penalty mechanism considers two fitness metrics: a convergence metric and a distribution metric.

Algorithm 2: Penalty Mechanism

```
Input: Solution Set: $\mathbf{F}$, Population Size: $n$
Output: Next Population: $\mathbf{P}$
Compute the distribution metrics of solutions $\mathbf{F}$: $\mathbf{DM}$;
Compute the convergence metrics of solutions $\mathbf{F}$: $\mathbf{CM}$;
$s = |\mathbf{F}|$;
Select a well-converged solution $p_i$ from $\mathbf{F}$;
Move $p_i$ into $\mathbf{P}$;
Distinguish the neighbors $Nei$ of $p_i$ based on $\mathbf{DM}$;
Move $Nei$ into the penalty set $\mathbf{E}$;
if $Size(\mathbf{E}) > s - n$ then
    Distinguish the neighbors $Nei$ of $\mathbf{P}$ based on $\mathbf{DM}$;
    penalize $Nei$ into the penalty set $\mathbf{E}$;
    Move others into the solution set $\mathbf{F}$;
end
```

For the convergence metric, the ideal strategy is to calculate the distance from the solution to the Pareto-optimal front of the problem [53]. However, in many cases, the Pareto-optimal front of the problem is unknown [54], i.e., we cannot directly use the above method to represent the convergence value. One of the improved methods uses the calculated distance from a solution to an ideal point (Z^L) as the solution's convergence value [46]. Here Z^L is composed of the minimum value of each objective for the current solutions. There are several distance metrics for measuring the convergence of solutions [28], [55], [56]. A distribution metric is used to detect the similarity between solutions in their convergence direction. For each solution, the connection between it and the ideal point is used to approximate its convergence direction. Tow solutions are similar if and only if their convergence directions are very close. The commonly used distribution metrics are the cosine metric [57] and the perpendicular distance [47]. In section III of the supplementary file, we carry out a detailed analysis of these metrics.

C. General Framework

The basic framework of the proposed algorithm is shown in Algorithm 3. First, an initial population $\mathbf{P}$ with n members is randomly constructed in the entire decision space. Second, in each iteration, n offspring solutions $\mathbf{S}$ are generated from the parent population by using the crossover operator [58] and the mutation operator [47]. Third, we select n solutions from the union set of $\mathbf{P}$ and $\mathbf{S}$ in each iteration. Finally, when the iteration is completed, this algorithm outputs the final obtained population.

During selection, we first divide solutions into different layers by the non-dominated sorting procedure. One non-dominated layer is selected at a time to construct a new population $\mathbf{P}$, starting from $\mathbf{F}_1$, until the size of $\mathbf{P}$ is equal to n or is for the first time greater than n. The last selected layer is defined as the critical layer $\mathbf{F}_l$. When the population is full ($|\mathbf{P}| = n$), $\mathbf{P}$ is returned. Conversely, if the population size exceeds the predefined size, we select partial solutions from $\mathbf{F}_l$ into the next generation of the population.

Algorithm 3: The basic framework of the algorithm

```
Input: Population Size: $n$, Terminate Condition: $\mathcal{T}$
Output: Next Population: $\mathbf{P}$
$\mathbf{P}$ = RandomInitiate($\mathbf{P}$);
while $\neg\mathcal{T}$ do
    $\mathbf{Q} = \emptyset$;
    $\mathbf{S}$ = MatingSelection($\mathbf{P}$);
    $\mathbf{S}$ = Crossover&Mutation($\mathbf{S}$);
    $\mathbf{R} = \mathbf{P} \cup \mathbf{S}$;
    $(\mathbf{F}_1, \mathbf{F}_2, \cdots)$ = Non-dominated-sorting($\mathbf{R}$);
    while $|\mathbf{F}| + |\mathbf{F}_i| \leqslant n$ do
        $\mathbf{P} = \mathbf{P} \cup \mathbf{F}_i$ and $i = i + 1$;
    end
    The last front to be included: $\mathbf{F}_l = \mathbf{F}_i$;
    if $|\mathbf{P}| = n$ then
        return $\mathbf{P}$;
    end
    $\overline{\mathbf{F}_l}$ = Pre-processing($\mathbf{F}_l$);
    Calculate the ratio of the spread of F and $\overline{\mathbf{F}_l}$ : r;
    if $\gamma > 1$ then
        $\mathbf{F}_l = \overline{\mathbf{F}_l}$;
    end
    Calculate the number of solutions $k$ to be punished
      from $\mathbf{F}_l$: $k = |\mathbf{P}| + |\mathbf{F}_l| - n$;
    Normalization($\mathbf{F}_l$);
    if $|\mathbf{P}| = 0$ then
        Select $m$ boundary solutions from $\mathbf{F}_l$, which
          are closest to the corresponding coordinate
          axes;
    end
    while $|\mathbf{P}| > n$ do
        Choose $n - |\mathbf{P}|$ solutions from $\mathbf{F}_l$:
          $\mathbf{P} = Penalty\_Mechanism(\mathbf{F}_l, n)$;
    end
end
```

Before selecting superior solutions from $\mathbf{F}_l$, the algorithm needs to detect if there are DRSs in $\mathbf{F}_l$. To this end, it first uses IQR to preprocess $\mathbf{F}_l$ to obtain a new set $\overline{\mathbf{F}_l}$. Afterwards, it compares the spreads of both sets ($\mathbf{F}_l$ and $\overline{\mathbf{F}_l}$). Spread indicator D is used to calculate the length of the diagonal of a hypercube formed by the extreme objective values attained in set $\mathbf{F}$, and is defined as

$$D = \left[\sum_{i=1}^{m} \left(\max_{\mathbf{x} \in \mathbf{F}} f_i(\mathbf{x}) - \min_{\mathbf{x} \in \mathbf{F}} f_i(\mathbf{x}) \right)^2 \right]^{1/2}, \tag{3}$$

where m denotes the number of objectives, and $\mathbf{x}$ is a candidate solution that belongs to $\mathbf{F}$. Finally, the ratio r of the spreads of the two sets is calculated by

$$\gamma = \frac{D_{\mathbf{F}_l}}{D_{\overline{\mathbf{F}_l}}}. \tag{4}$$

Obviously, if γ is smaller than or equal to 1, it means that there is no DRS in $\mathbf{F}_l$. On the contrary, if γ is greater than 1, it indicates that one or more DRSs exist in $\mathbf{F}_l$, and we consider $\mathbf{F}_l = \overline{\mathbf{F}_l}$. The influence of the preprocessing operation on our algorithm is considered in section IV the supplement file.

After preprocessing, the min-max normalization is used. The role of normalization is to scale the objectives of solutions into the same interval according to their ranges. Formally

$$f_i'(\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) - Z_i^L}{Z_i^U - Z_i^L}, \quad i \in \{1, 2, \cdots, m\}, \tag{5}$$

where the ideal point Z^L and the nadir point Z^U are determined by figuring out the minimum and maximum, respectively, values of each objective for all solutions in the preprocessed set.

Once the normalization operation has been completed, two cases are considered. Since all of the solutions are located on the unified layer, PMEA first prefers to select m boundary solutions to ensure the spread of the population and subsequently uses the penalty mechanism. Otherwise, it selects $n-|\mathbf{P}|$ well-converged solutions one-by-one from the preprocessed set by the penalty mechanism.

The above description implies that the computational complexity of the proposed algorithm in one generation is dominated by the non-dominated sorting and the penalty mechanism. Specifically, the non-dominated sorting of a population with $2n$ solutions and m objectives requires $O(n(\log n)^{m-2})$ computations [59]. Preprocessing solutions requires $O(n \log n)$ computations. Normalization of objectives requires $O(mn)$ computations. In the penalty mechanism, the computational complexities of the calculations of cosine similarity and elitism selection are $O(mn^2)$ and $O(n^2)$, respectively. Considering all the above computations, the overall worst-case complexity of one generation of this algorithm is $O(mn^2)$.

IV. Results and Analysis

In this section, the performance of PMEA is empirically evaluated by comparison with various types of state-of-the-art algorithms, such as SPEA2+SDE [23], NSGA-II+SDR [60], MaOEAIGD [61], VaEA [45], SPEAR [62]. The experiments are conducted on 25 test problems taken from three widely used test suites, namely, DTLZ [52], WFG [63] and MaF [64]. For each test problem, 3, 5, 8, 10 and 15 objectives are considered. In addition, two widely-used performance indicators, namely, the Hypervolume (HV) [65] and inverted generational distance (IGD) [66], are considered. Due to the page limit, the experimental design, the statistical results are described in the supplementary file. In particular, the Section V in the supplementary file gives the experimental settings of this paper and the Section VIII in the supplementary file gives the statistical results.

For comparison purposes, we use the Manhattan distance and the cosine distance as the convergence and distribution metrics of our algorithm (named PMEA-MA). In section VI of the supplementary file, we demonstrate that the combination of the Manhattan distance and the cosine distance can provide the best performance for PMEA. Furthermore, in the following experiments, we also compare PMEA-MA with a variant of PMEA-MA without preprocessing (named PMEA*-MA)

TABLE II. THE PROPORTIONS OF THREE TEST SUITES ON WHICH PMEA-MA ARE BETTER THAN, WORSE THAN, AND EQUAL TO SIX PEER ALGORITHMS CONCERNING IGD INDICATOR

MPEA-MA v.s.		PMEA*-MA	SPEA2+SDE	NSGA-II+SDR	MaOEAIGD	VaEA	SPEAR
DTLZ1	+	2/35	16/35	8/35	2/35	3/35	3/35
	−	8/35	19/35	27/35	33/35	28/35	32/35
	≈	25/35	0/35	0/35	0/35	4/35	0/35
WFG	+	10/45	12/45	11/45	0/45	9/45	13/45
	−	1/45	32/45	33/45	44/45	31/45	30/45
	≈	34/45	1/45	1/45	1/45	5/45	2/45
MaF	+	2/45	16/45	5/45	3/45	8/45	2/45
	−	14/35	28/45	40/45	42/45	8/45	42/45
	≈	29/45	1/45	0/45	0/45	7/45	1/45

The symbols "+", "≈" and "−" denote that the performance of the compared algorithm is statistically better than, equivalent to, and worse than PMEA-MA.

to assess the effectiveness of preprocessing and the penalty mechanism.

A. DTLZ problems

Table IV of the supplementary file presents a comparison of results of the seven algorithms on the DTLZ test suite. It is apparent that PMEA-MA and SPEA2+SDE have a clear advantage over the other five algorithms on the majority of the test instances. Our algorithm is highly competitive on problems with concave Pareto fronts, i.e., DTLZ2, DTLZ3 and DTLZ4, while SPEA2+SDE is good at solving problems with simple Pareto fronts, i.e., DTLZ1, and problems with degenerate Pareto fronts, i.e., DTLZ5 and DTLZ6. These two algorithms obtain 16 and 13, repsectively, best results in the entire DTLZ test suite. NSGA-II+SDR outperforms our algorithm only when solving problems with degenerate Pareto fronts. MOEAIGD does not obtain the best results on any problem the entire DTLZ suite and only outperforms our algorithm when optimizing 8-objective DTLZ2 and 15-objective DTLZ7 problems. VaEA only obtains the best result on the 8-objective DTLZ7 problem. Although VaEA also uses the Manhattan distance and the cosine distance to represent the solutions' convergence and distribution metrics, its performance is far from that of our algorithm. The main reason is that VaEA places too much emphasis on diversity and is weak on convergence. Matching the performance of NSGA-II+SDR, SPEAR only obtains two best results among 35 problems. Additionally, the IGD values of PMEA-MA and PMEA*-MA demonstrate that the preprocessing operation can improve the performance of our algorithm when dealing with DTLZ problems.

As Fig. 20 of the supplementary file shows, the approximate Pareto front obtained by PMEA-MA on DTLZ3 exhibits promising convergence and good diversity. Although the solutions obtained by SPEA2+SDE have good convergence, their distribution is not as uniform as that of solutions obtained by our algorithm. For NSGA-II+SDR, the final solution set obtained by optimizing the 10-objective DTLZ3 problem is much better than that obtained by solving the 3-objective DTLZ3 problem. Because MaOEAIGD can easily fall into a local optimum when solving the DTLZ3 problem, its performance is not as good as that of our algorithm. Conversely, VaEA and SPEAR perform better on low-dimensional DTLZ3 problems, but lack effective convergence on high-dimensional DTLZ3 problems.

In short, our algorithm is more competitive than the compared algorithms on DTLZ problems. As for the statistical results of Table II, the proportions of the DTLZ test instances on which PMEA-MA outperforms PMEA*-MA, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA and SPEAR with statistical significance are 8/35, 19/35, 27/35, 33/35, 28/35 and 32/35, respectively. Conversely, the proportions of the DTLZ test instances on which PMEA-MA outperforms PMEA*-MA, BiGE, NSGAIII, KnEA, VaEA and SPEAR with statistical significance are 2/35, 16/35, 8/35, 2/35, 3/35 and 3/35, respectively.

B. WFG problems

Table V of the supplementary file presents the IGD results for all WFG test instances. As the table shows, PMEA-MA has an absolute advantage on problems with concave Pareto fronts, such as WFG4-9. However, PMEA-MA is not as effective as other algorithms on problems with complex Pareto fronts, such as WFG1-3. If the preprocessing operation is removed, we observe that the performance of our algorithm on these problems improves. Compared to other algorithms, SPEA2+SDE is very competitive on the WFG1 problem. However, it is easy to note that the performance of SPEA2+SDE on the WFG test suite is not as good as that on the DTLZ test suite. The main reason is that SDE is likely to fail to reflect the solutions' performance in the case of the scaled problems. NSGA-II+SDR, only obtains the best results on two problems, namely, 5- and 8-objective WFG3. MaOEAIGD does not perform as well as other algorithms on the WFG test suite. VaEA performs better on the WFG suite than the DTLZ suite. The reason is that WFG problems emphasize diversity rather than convergence, while VaEA mainly uses a distributed maintenance mechanism to select solutions. SPEAR performs well on WFG test problems in a low-dimensional space. The multilayer simplex lattice approach can only alleviate but not completely solve the problem of setting weights in a high-dimensional space. In addition, the table shows that preprocessing has less impact on most of the WFG problems.

Fig. 21 of the supplementary file presents the final solutions of one run with respect to the 10-objective WFG9 problem obtained by parallel coordinates. For this problem, the range of the ith objective is $[0, 2 \times i]$. As shown, the solutions obtained by these algorithms are similar in convergence while slightly different in diversity. Specifically, PMEA-MA and VaEA perform the best, and obtain solutions that can even be distributed over the entire Pareto front. The solution set obtained by SEPA2+SDE is biased toward the regions far away from the ideal point. Although the solutions obtained by NSGA-II+SDR cover the entire objective space, their distribution is not ideal. The solutions obtained by MaOEAIGD converge to several local regions. The solutions obtained by SPEAR are also distributed evenly, while the boundary solutions seem to be denser than the internal ones

In summary, we can conclude from above descriptions that PMEA-MA performs the best among the five compared algorithms on WFG problems. As shown in Table II, the proportions of WFG test instances on which PMEA-MA outperforms PMEA*-MA, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA and SPEAR are 1/45, 32/45, 33/45, 44/45 31/45 and 30/45, respectively. Conversely, the proportions of test isntances on which PMEA-MA is outperformed by the compared algorithms are 10/45, 12/45, 11/45, 0/45, 9/45 and 13/45 for PMEA*-MA, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA and SPEAR, respectively. It must be acknowledged that preprocessing indeed has a detrimental effect on our algorithm when optimizing WFG1 and WFG2 problems. For those problems, most solutions prefer the regions that are close to the ideal point, causing the preprocessing operation to remove the sparse solutions of the regions far from the ideal points as abnormal solutions.

C. MaF problems

The IGD results for MaF test instances are given in Table VI of the supplementary file. As the table shows, PMEA-MA performs the best, exhibiting a clear advantage over the other five algorithms on the majority of test instances. More specifically, PMEA-MA obtains the best and second-best IGD results on 37 out of 45 test instances, while other algorithms obtain the best and second-best IGD results on 19, 16, 4, 0, 12 and 2 out of 45 test instances for PMEA*-MA, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA and SPEAR, respectively. PMEA-MA is more suitable than the other compared algorithms for MaF1 with an inverted linear Pareto front and defeats opponents in all dimensions of this problem. For MaF2 with a concave Pareto front, PMEA-MA performs the best on low-dimensional instances while SPEA2+SDE and VaEA perform the best on high-dimensional instances. The MaF3 problem mainly examines whether a MOEA handles convex and multimodal Pareto front's shape. In this problem, SPEA2+SDE clearly outperforms the other six algorithms and is followed by our algorithm. MaF4 and MaF5 are designed to investigate the ability of algorithms to handle problems with different scaled objective values. PMEA-MA and VaEA perform very well on such problems. The former performs the best on MaF5, and the latter outperforms the other algorithms on most of MaF4 instances. The Pareto front of MaF6 is a degenerate curve designed to test the ability of algorithms to find a lower-dimensional objective space. In this problem, the performance of our algorithm does not seem to be as good as that of the other algorithms, i.e., NSGA-II+SDR, MaOEAIGD and VaEA, especially in high-dimensional objective spaces. MaF8 and MaF9 consider a two-dimensional decision space and calculate the Euclidean distance from points to M target points and target straight lines of a given polygon, respectively. In these two problems, PMEA-MA and PMEA*-MA outperform the other compared algorithms on MaF8, while SPEA2+SDE obtains the best IGD values on MaF9 problems of any dimension. MaF13 is designed primarily to verify the convergence of an algorithm. The data show that PMEA-MA is the best performer in most cases.

For further observations, Fig. 22 of the supplementary file plots the final solution set obtained by each algorithm on MaF8 problems with 10 objectives. The Pareto-optimal solutions of this problem are designed to be inside one (or several) two-dimensional closure(s) in the decision space. The centre coordinates of the Pareto-optimal region are (0,0), and the radius of the Pareto-optimal region is 1.0. This figure shows that PMEA-MA has the bese diversity and is followed by VaEA. The difference between the two algorithms is that the convergence of the former is better than that of the latter. For SPEA2+SDE, NSGA-II+SDR and MaOEAIGD, although most of their solutions converge to the Pareto front, the diversity of those solutions is not ideal. More specifically, the solutions obtained by MaOEAIGD converge to local regions in this problem. In contrast, SPEAR has many badly converged solutions, most of which are concentrated in several small regions.

Overall, PMEA-MA also performs very well on MaF problems. As the statistical results in Table II, PMEA-MA shows significant improvement over other algorithms on most MaF instances. The proportions of test instances on which PMEA-MA outperforms better than PMEA*-MA, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA, and SPEAR are 14/45, 29/45, 40/45, 42/45, 30/45 and 42/45, respectively.

V. Conclusions

The "curse of dimensionality" poses great challenges for traditional Pareto-based algorithms. The imbalance between the convergence and diversity in many-objective optimization suggests the need for new methodologies in evolutionary computation. In this paper, we propose a novel Pareto-based evolutionary algorithm with a penalty mechanism to deal with many-objective problems. By alternating the selection and penalty operations, PMEA considers both the convergence and diversity for each solution in the population. In addition, to eliminate the outliers and improve the stability of PMEA, we introduce a preprocessing mechanism before the selection operation.

In fact, the penalty mechanism is part of a selection framework in which any kind of convergence metrics and distribution metrics can be used as relevant information. In this paper, we have considered several common distance metrics (e.g., the Manhattan distance, the Euclidean distance and the Chebyshev distance) and distribution metrics (e.g., the cosine metric and the perpendicular distance). The experimental results show that the combination of the Manhattan distance and the cosine metric is the best. In addition, experiments comparing the proposed method with eight popular MOEAs (namely, SPEA2+SDE, NSGA-II+SDR, MaOEAIGD, VaEA, SPEAR) demonstrate that PMEA-MA significantly outperforms its competitors on most problem instances.

Future studies will include further exploring the potential of the proposed algorithm. In this paper, we have considered the same neighbourhood range for all solutions in the population, which may not be conducive to solving problems with complex Pareto fronts (e.g., degenerate problems). Therefore, adapting the neighbourhood range to the population during the search could be one possible avenue for improving the algorithm's performance.

References

[1] Y. Xu, K. Li, J. Hu, and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues," *Inf. Sci.*, vol. 270, pp. 255–287, Jun. 2014.

[2] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1344–1357, May 2016.

[3] K. Li, X. Tang, and K. Li, "Energy-efficient stochastic task scheduling on heterogeneous computing systems," *IEEE Trans. on Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2867–2876, Nov. 2014.

[4] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, "Hyperplane assisted evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Cybern.*, early access, Mar. 4, 2019, doi: 10.1109/TCYB.2019.2899225.

[5] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 145–160, Feb. 2016.

[6] H. Xu, W. Zeng, X. Zeng, and G. G. Yen, "An evolutionary algorithm based on minkowski distance for many-objective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3968–3979, Nov. 2019.

[7] M. Li and X. Yao. (2017) What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multi-objective optimisation. [Online]. Available: arxiv.abs/1709.02679

[8] Z. He and G. G. Yen, "Many-objective evolutionary algorithms based on coordinated selection strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220–233, Apr. 2017.

[9] P. Wang, B. Liao, W. Zhu, L. Cai, S. Ren, M. Chen, Z. Li, and K. Li, "Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D," *Appl. Soft Comput.*, vol. 70, pp. 797–813, Sep. 2018.

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

[11] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.

[12] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Germany: Springer, 2007, pp. 5–20.

[13] M. Farina and P. Amato, "A fuzzy definition of 'optimality' for many-criteria optimization problems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 3, pp. 315–326, May 2004.

[14] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Germany: Springer, 2005, pp. 399–412.

[15] H. Wang and X. Yao, "Corner sort for pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan. 2014.

[16] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[17] Y. Liu, N. Zhu, K. Li, M. Li, J. Zheng, and K. Li, "An angle dominance criterion for evolutionary many-objective optimization," *Inf. Sci.*, vol. 509, pp. 376–399, Jan. 2020.

[18] K. Narukawa, "Effect of dominance balance in many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Berlin, Germany: Springer, 2013, pp. 276–290.

[19] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924–938, Dec. 2016.

[20] Y. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, and K. Shang, "Improving 1by1EA to handle various shapes of pareto fronts," in *Parallel Problem Solving from Nature–PPSN XV*, A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley, Eds. Cham, Switzerland: Springer Int., 2018, pp. 311–322.

[21] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Germany: Springer, 2007, pp. 742–756.

[22] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.

[23] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[24] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Survs.*, vol. 52, no. 2, pp. 1–38, Mar. 2019.

[25] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature–PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, 2004, pp. 832–842.

[26] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Germany: Springer, 2005, pp. 62–76.

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

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

[29] Y. Zhou, Y. Xiang, Z. Chen, J. He, and J. Wang, "A scalar projection and angle-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2073–2084, Jun. 2019.

[30] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3–18, Feb. 2018.

[31] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[32] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.

[33] H. Jain and K. Deb, "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.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

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

[35] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–33, Jul. 2013.

[36] R. K. Ursem, "Diversity-guided evolutionary algorithms," in *Parallel Problem Solving from Nature–PPSN VII*, J. J. M. Guervós, P. Adamidis, H.-G. Beyer, H.-P. Schwefel, and J.-L. Fernández-Villacañas, Eds. Berlin, Germany: Springer, 2002, pp. 462–471.

[37] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in GA," *Appl. Soft Comput.*, vol. 24, pp. 1047–1077, 2014.

[38] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of pareto-based MOEAs:

does non-dominated really mean near to optimal?" in *Proc. Congr. Evol. Comput.*, vol. 2, May 2001, pp. 957–962.

[39] G. Yu, R. Shen, J. Zheng, M. Li, J. Zou, and Y. Liu, "Binary search based boundary elimination selection in many-objective evolutionary optimization," *Appl. Soft Comput.*, vol. 60, pp. 689–705, 2017.

[40] K. S. Bhattacharjee, H. K. Singh, T. Ray, and Q. Zhang, "Decomposition based evolutionary algorithm with a dual set of reference vectors," in *IEEE Congr. Evol. Comput. (CEC)*. IEEE, Jun. 2017, pp. 105–112.

[41] W. Hu, G. G. Yen, and G. Luo, "Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1446–1459, Jun. 2017.

[42] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[43] M. Zhang and H. Li, "A reference direction and entropy based evolutionary algorithm for many-objective optimization," *Appl. Soft Comput.*, vol. 70, pp. 108–130, Sep. 2018.

[44] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, p. 13, 2015.

[45] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, Feb. 2017.

[46] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, Sep. 2017.

[47] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[48] Q. Zhu, Q. Lin, J. Li, C. A. Coello Coello, Z. Ming, J. Chen, and J. Zhang, "An elite gene guided reproduction operator for many-objective optimization," *IEEE Trans. Cybern.*, pp. 1–14, 2019, doi: 10.1109/TCYB.2019.2932451.

[49] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.

[50] R. Shen, J. Zheng, M. Li, and J. Zou, "Many-objective optimization based on information separation and neighbor punishment selection," *Soft Comput.*, vol. 21, no. 5, pp. 1109–1128, 2017.

[51] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Min.*, vol. 10, no. 1, p. 35, Dec. 2017.

[52] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005, pp. 105–145.

[53] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a pareto front," in *Proc. Genet. Program. Conf.*, 1998, pp. 221–228.

[54] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, "Guiding evolutionary multiobjective optimization with generic front modeling," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1106–1119, Mar. 2020.

[55] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45–65, Nov. 2015.

[56] W.-C. Chang, A. Sutcliffe, and R. Neville, "A distance function-based multi-objective evolutionary algorithm," in *Proc. Genet. Evol. Comput. Conf.* Citeseer, 2003, pp. 47–53.

[57] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1216–1230, Jul. 2012.

[58] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[59] M. T. Jensen, "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms," *IEEE Trans. on Evol. Comput.*, vol. 7, no. 5, pp. 503–515, Oct. 2003.

[60] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 331–345, Apr. 2019.

[61] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 173–187, Apr. 2019.

[62] S. Jiang and S. Yang, "A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, Jun. 2017.

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

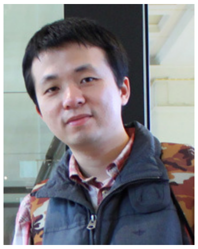
[64] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 1, pp. 67–81, Mar. 2017.

[65] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

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

Yuan Liu received the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University, Xiangtan, China, in 2017. He is currently working toward the Ph.D. degree with the College of Information Science and Engineering, Hunan University, Changsha, China. His current research topic is many-objective optimization.

Ningbo Zhu received the Ph.D. degree in engineering from the computer use and application, Nanjing University of Science and Technology, Nanjing, China, in 2005. He is a Associate Professor at the College of Information Science and Engineering, Hunan University, Changsha, China. His current research interests include pattern recognition and intelligent system, digital image processing and evolutionary computation.



Miqing Li received the Ph.D. degree in computer science from the Department of Computer Science, Brunel University London, U.K. in 2015.

Dr Miqing Li is currently a lecturer with the University of Birmingham, U.K. His research is principally on multi-objective optimisation, where he works on developing population-based randomised algorithms (e.g. evolutionary algorithms) for both general challenging problems (e.g. many-objective optimisation, constrained optimisation, multi-modal optimisation, robust optimisation, expensive optimisation) and specific challenging problems in other fields (e.g. software engineering, system engineering, product disassembly, post-disaster response, neural architecture search, and reinforcement learning). He is the founding chair of IEEE CIS Task Force on Many-Objective Optimisation.