

# A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy

Yiping Liu, Dunwei Gong, Jing Sun, and Yaochu Jin, *Fellow, IEEE*

**Abstract**—Most existing multi-objective evolutionary algorithms experience difficulties in solving many-objective optimization problems due to their incapability to balance convergence and diversity in the high-dimensional objective space. In this paper, we propose a novel many-objective evolutionary algorithm using a one-by-one selection strategy. The main idea is that in the environmental selection, offspring individuals are selected one by one based on a computationally efficient convergence indicator to increase the selection pressure towards the Pareto optimal front. In the one-by-one selection, once an individual is selected, its neighbors are de-emphasized using a niche technique to guarantee the diversity of the population, in which the similarity between individuals is evaluated by means of a distribution indicator. In addition, different methods for calculating the convergence indicator are examined and an angle-based similarity measure is adopted for effective evaluations of the distribution of solutions in the high-dimensional objective space. Moreover, corner solutions are utilized to enhance the spread of the solutions and to deal with scaled optimization problems. The proposed algorithm is empirically compared with eight state-of-the-art many-objective evolutionary algorithms on 90 instances of 16 benchmark problems. The comparative results demonstrate that the overall performance of the proposed algorithm is superior to the compared algorithms on the optimization problems studied in this work.

**Index Terms**—Many-objective optimization, evolutionary multi-objective optimization, performance indicator, cosine similarity, convergence, diversity.

## I. INTRODUCTION

MULTI-objective optimization problems (MOPs) are commonly seen in real world applications, e.g., industrial scheduling [1], controller design [2], [3], environmental/economic dispatch [4], and many others. MOPs are characterized with multiple conflicting objectives, indicating that there is no single optimal solution to these problems, rather a set of trade-off solutions, known as the Pareto optimal set. In the literature, MOPs with more than three objectives are often referred to as many-objective optimization problems (MaOPs) [5]–[9].

Y. Liu is with School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China. e-mail: lyp.cumt@gmail.com

D. Gong is with School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China, and School of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China. e-mail: dwgong@vip.163.com

J. Sun is with School of Science, Huai Hai Institute of Technology, Lianyungang 222005, China. e-mail: jing8880@sina.com

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom. e-mail: yaochu.jin@surrey.ac.uk

Manuscript received 2015; revised 2015.

Over the past two decades, a large number of multi-objective evolutionary algorithms (MOEAs) have been proposed. Generally speaking, the selection strategy plays a key role in most MOEAs, which needs to take both the convergence of the solutions to the Pareto optimal front and the distribution of the solutions into account.

Pareto-dominance-based MOEAs, e.g., nondominated sorting genetic algorithm II (NSGA-II) [10] and strength Pareto evolutionary algorithm 2 (SPEA2) [11], are the most popular among various methods. These algorithms employ the Pareto dominance as the primary selection criterion, where non-dominated solutions are always preferred for good convergence. To promote a good distribution of the solutions, a diversity related secondary selection criterion is adopted as well. Dominance-based MOEAs have been proved successful in solving a large number of MOPs. However, their performance will seriously deteriorate in solving MaOPs. One main reason is that the proportion of non-dominated solutions in the population considerably rises as the number of objectives increases [12], leaving the dominance-based selection criterion unable to distinguish solutions. As a result, the diversity based criterion will play a crucial role in selection, leading to solutions far away from the Pareto optimal front being maintained [13].

An effective approach to improving the ability of dominance-based MOEAs to solve MaOPs is to redefine the dominance relationship. A large body of research along this line has been reported, e.g., dominance area control [14], L-optimality [15], fuzzy Pareto dominance [16], etc. Although these modified dominance relationships have been shown to be able to improve the convergence of MOEAs for solving MaOPs, they may also cause the population to converge to a sub-region of the Pareto optimal front [17].

An alternative way to enhance the ability of dominance-based MOEAs for MaOPs is either to improve the diversity-based secondary selection criterion, or to replace it with other selection criteria. For instance, a diversity management operator is employed to adjust the requirement on diversity in mating and environmental selection [18]. By shifting the position of a solution in sparse regions, a shift-based density estimation (SDE) was proposed to discard solutions that lead to poor convergence [19]. An extension of NSGA-II for solving MaOPs, termed NSGA-III was developed in [20], which uses a reference-point-based secondary criterion for maintaining diversity. In [21], following the idea of preferring the knee points among non-dominated solutions, a knee point driven evolutionary algorithm (KnEA) was proposed for many-objective optimization.

Unlike dominance-based MOEAs, decomposition-based MOEAs have been found to be very promising for many-objective optimization. One typical example of this class of algorithms are MOEA/D [22] and its variants [23]–[27]. In MOEA/D, the diversity of a population is maintained by a set of predefined well distributed reference vectors. Based on these reference vectors, the objectives of an MOP are aggregated into a number of scalarizing functions, each of which generating a single scalar value. Individuals in a population are guided to search towards the Pareto optimal front in the directions specified by the reference vectors by minimizing these scalarizing functions. MOEA/D has been demonstrated efficient in solving both MOPs and MaOPs. However, one known problem of MOEA/D is that uniformly distributed reference vectors do not necessarily lead to uniformly distributed solutions, particularly for problems with irregular (i.e., nonuniform) Pareto optimal fronts.

Grid-based algorithms, e.g.,  $\epsilon$ -MOEA [28], GrEA [29], and MaOPSO/2s-pccs [30], are also very competitive MOEAs for solving MaOPs. Grids are very effective for measuring the distribution of solutions according to their positions. In these algorithms, the objective space is first divided into a grid of hypercubes. Then, some convergence-related selection criteria, e.g., the dominance relationship of the hypercubes and the distance between a solution and the origin of its hypercube, are utilized so that each non-dominated hypercube contains no more than one solution. However, the adaptation of the grid size for different optimization problems is non-trivial.

Different from the above methods, indicator-based evolutionary algorithms (IBEAs) [31] adopt a single indicator which accounts for both convergence and distribution performances of a solution set. Therefore, solutions can be selected one by one based on their influence on the performance indicator. Among others, hypervolume is a widely used indicator in IBEAs. Unfortunately, the computational complexity for calculating hypervolume increases exponentially as the number of objectives increases, which makes it computationally prohibitive for MaOPs. To address this issue, HypE [32] uses Monte Carlo simulations to estimate the hypervolume, where the accuracy of the estimated hypervolume can be trade off against the available computational resources. Recently, R2 [33] and the additive approximation [34] were also proposed to further enhance the computational efficiency of IBEAs for solving MaOPs.

The effectiveness of IBEAs for solving MaOPs lies in the fact that performance indicators such as hypervolume can account for both convergence and distribution of the solution set. Intuitively, it is also likely to consider convergence and distribution of a solution set separately, e.g., ~~by simultaneously measuring the distance of the solutions to the Pareto optimal front, and maintaining a sufficient distance between each other.~~ Motivated by this idea, we propose here a many objective evolutionary algorithm using a one-by-one selection strategy, ~~1by1EA for short.~~ In 1by1EA, solutions are selected according to a convergence indicator and a distribution indicator. The former measures the distance between a solution and the Pareto optimal front, while the latter measures its distance to each other. In each step of the selection procedure, only

one solution with the best value of the convergence indicator is selected. Then, based on the distribution indicator, the solutions close to the selected one are de-emphasized using a niche technique. Thus, superior solutions are selected one by one and the needed calculation in each generation is little. On the contrary, calculating the indicator in IBEAs is computationally expensive, not only because the calculation involves both convergence and distribution, but also because it has to be recalculated once a new solution is selected.

The main contributions of this work can be summarized as follows:

(1) A general evolutionary framework, termed 1by1EA is presented, where solutions in the current population are selected one by one based on a convergence indicator. The neighboring solutions of the selected solution are de-emphasized using a niche technique, in which their similarities are evaluated by a distribution indicator. One benefit of this selection strategy is that it is able to easily balance the need for convergence towards the Pareto optimal front and the requirement on diversity of the population.

(2) A distribution indicator based on the cosine similarity is proposed for effectively evaluating the distance between solutions in the high-dimensional objective space. Compared with the distribution indicator based on the Euclidean distance, cosine similarity can effectively reduce the number for solutions that prevent the population from converging to the Pareto optimal front, thereby enhancing the algorithm's ability to solve MaOPs.

(3) A boundary maintenance mechanism is suggested by taking advantage of the corner solutions to achieve a good spread of the solutions over the Pareto optimal front. We demonstrate that this mechanism can reduce the possibility of losing boundary solutions.

The remainder of this paper is organized as follows. In Section II, important definitions in multi-objective optimization are given and the motivation of this work is elaborated. The proposed 1by1EA is then described in detail in Section III. Section IV presents the experimental design, test functions, and performance indicators for comparing the performance of 1by1EA with the state-of-the-art. The experimental results and discussions are given in Section V. Section VI concludes the paper.

## II. PRELIMINARIES

### A. Basic Definitions

Without loss of generality, the MOP considered in this study is formulated as follows:

$$\begin{aligned} \min & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s.t. } & \mathbf{x} \in S \subset \mathbf{R}^n \end{aligned} \quad (1)$$

where  $\mathbf{x}$  represents an  $n$ -dimensional decision vector in space  $S$ .  $f_m(\mathbf{x})$ ,  $m = 1, 2, \dots, M$  is the  $m$ -th objective to be minimized, and  $M$  is the number of objectives. When  $M > 3$ , this problem is known as an MaOP.

In multi-objective optimization, the following concepts have been well defined and widely applied.

**Pareto Dominance:** For any two different solutions of formula (1),  $\mathbf{x}_1, \mathbf{x}_2 \in S$ , if  $\forall m = 1, 2, \dots, M, f_m(\mathbf{x}_1) \leq f_m(\mathbf{x}_2)$ ,

and  $\exists i = 1, 2, \dots, M, f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$ , then  $\mathbf{x}_1$  dominates  $\mathbf{x}_2$ , denoted as  $\mathbf{x}_1 \prec \mathbf{x}_2$ .

**Pareto optimal set:** For a solution to the problem in (1),  $\mathbf{x}^* \in S$ , if there is no  $\mathbf{x}' \in S$  satisfying  $\mathbf{x}' \prec \mathbf{x}^*$ , then  $\mathbf{x}^*$  is the Pareto optimal solution. All such solutions form a set often called the Pareto optimal solution set ( $PS$ ).

**Pareto optimal front:** The image of the Pareto optimal solution set in the objective space is known as the Pareto optimal front ( $PF$ ).

## B. Motivation

One major reason for MOEAs to have difficulties in solving MaOPs is that it is no longer possible to distinguish solutions by means of the Pareto dominance relationship [12]. Many efforts have been made to modify the dominance relationship to make it easier to distinguish solutions [14]–[16]. In addition, it will be much easier to distinguish solutions if we utilize a scalarizing function to measure their convergence performance. As a result, superior solutions can be selected one by one, and the selection pressure towards the Pareto optimal front will be significantly increased. Motivated by this idea, we adopt a convergence indicator that sums up the values of all objectives of a solution. This convergence indicator is somewhat similar to the scalarizing function in MOEA/D [22], or the achievement function in preference-based MOEAs [35]. Unlike these methods, however, no reference/weight vectors are predefined for the objectives, since all objectives are considered equally important. In this way, an MaOP is actually transformed into a single-objective problem whose objective is the convergence indicator.

Note however that if we adopt the convergence performance for selecting solutions, only a small part of Pareto optimal front can be obtained. To address this issue, a distribution measure is needed. As we know, niche techniques have widely been used in handling multi-modal single-objective optimization problems due to their ability in maintaining the diversity of a population in the decision space [36], [37]. In fact, niche techniques have also been used to maintain the diversity in multi-objective optimization [38], which was termed niched Pareto genetic algorithm (NPGA). In NPGA, a niche is determined by a threshold, and the niche count is the number of solutions in a niche. Non-dominated solutions with a smaller niche count are preferred. Deb et al. [39] proposed a similar method called  $\varepsilon$ -clearing, in which non-dominated solutions are randomly selected, and then their neighbors are de-emphasized. In [40], an MaOP is transformed into a bi-objective problem, in which one objective is constructed based on a niche technique. Optimizing this objective leads to a more diverse population. Recently, Zhang et al. [21] employed a hyperbox to identify the neighbors of a selected solution, and the size of the hyperbox can be adaptively changed during the evolution. Inspired by the above ideas, in this work, we adopt a distribution indicator, which is a vector with each of its elements representing the distance between a solution and the rest ones. Once a solution is selected, all solutions whose distance to the selected one is less than a predefined threshold will be de-emphasized.

Using the above convergence and distribution indicators as two selection criteria, the evolutionary algorithm will be able to solve MaOPs in principle. However, the following issues must be resolved to enhance the performance:

(1) Solutions selected according to the above criteria may contain dominated solutions. Assume there are nine solutions in a population, solutions A to I, for a bi-objective optimization problem, as illustrated in Fig. 1. Suppose that the convergence indicator is the Euclidean distance of a solution to the origin, and the distribution indicator is a vector of the Euclidean distances between this solution to the rest. Clearly, if a solution dominates another, it is very likely to have a better value in terms of the convergence indicator. Assume  $r_\alpha$  is the distance threshold to de-emphasize the neighboring solutions once a solution is selected. For the given solutions, A is the first solution to be selected according to the convergence indicator. Then, according to the distribution indicator and the value of  $r_\alpha$ , B is de-emphasized. Solution C, which is closer to the origin than solutions D to I, is selected thereafter, although it is dominated by A. By contrast, solution D or F, which are non-dominated by solution A, will not be selected. This means that the selection criterion is not strictly consistent with the Pareto dominance criterion. Although the probability that one solution dominates another is considerably low in a high-dimensional objective space, selection of dominated solutions instead of the non-dominated ones may degrade the search performance of the algorithm.

(2) Euclidean distance is not well suited for measuring distribution in a high-dimensional space. Most existing methods, however, such as niche [38] and  $k$ -nearest [11], employ the Euclidean distance to estimate the distribution of solutions, which results in a large number of dominance resistant solutions ( $DRS$ ) [13] in the population. For example in Fig. 1, F-I are all selected since for each of them, the Euclidean distance to each of the others is larger than the threshold. As a matter of fact, G, H, and I are farther away from the Pareto optimal front than some solutions which are in a more crowded region, e.g. B. This issue becomes more serious in a high-dimensional space, since the Euclidean distance has been shown to be inadequate to describe the neighborhood relationship in a high-dimensional space [41].

(3) Boundary solutions may be lost during the evolution. For instance, in Fig. 1, boundary solution E is lost once solution D is selected.

(4) An effective method for determining the right threshold still lacks, which is critical for achieving a good distribution performance. Zhang et al. [21] proposed an adaptive strategy based on information from the population at the previous generation. However, information about the shape of the non-dominated front in the current and previous populations may be misleading before the shape of the true Pareto optimal front becomes clear.

To implement the ideas and resolve the issues discussed above, we propose here a many-objective evolutionary algorithm using a one-by-one selection strategy. In the proposed method, individuals (candidate solutions) are first sorted according to a convergence indicator. In each step, only the individual having the best convergence performance will be

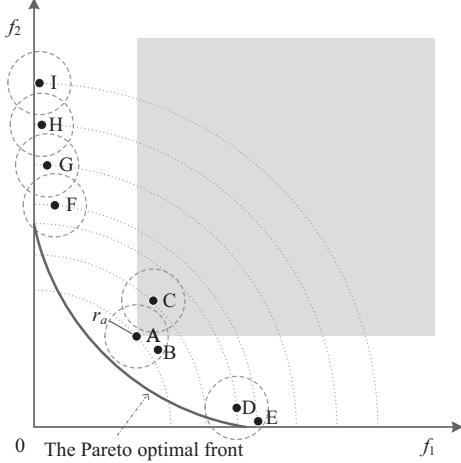


Fig. 1. An illustration of selecting solutions one by one for a bi-objective problem. In this example, dots A to I indicate solutions in the current population. In this example, the convergence indicator of a solution is its Euclidean distance to the origin, and the distribution indicator is its Euclidean distances to other solutions. Solutions on the arcs denoted by a dotted line have the same value in terms of the convergence indicator.  $r_\alpha$  is a threshold to de-emphasize neighboring solutions, and the circles denoted by dashed lines are the niches formed by  $r_\alpha$ . On the basis of the convergence and the distribution indicators, solutions A, C, D, F, G, H and I are selected one by one, and B and E are de-emphasized after A and D are selected, respectively. In addition, the shaded area indicates the dominant region dominated by solution A.

selected. Then, individuals in the neighborhood of the selected one or dominated by the selected one are de-emphasized. To efficiently evaluate the distance between solutions in a high-dimensional space, a distribution indicator based on the cosine similarity is employed. In addition, a boundary maintenance mechanism and a normalization method based on the corner solutions are introduced to ensure a good coverage of the Pareto optimal front and to enhance the algorithm's ability in solving scaled problems. The detail of 1by1EA will be presented in Section III.

It is worth noting that the knee point driven environmental selection proposed in KnEA [21] can also be seen as a one-by-one selection strategy, however, it is used as a secondary selection criterion in addition to non-dominated sorting proposed in NSGA-II [10]. In this work we attempt to present a general framework that utilizes the convergence indicator as the primary selection criterion.

### III. THE PROPOSED METHOD

#### A. A General Framework

Algorithm 1 presents the overall framework of the proposed 1by1EA, which is similar to most generational MOEAs. First, an initial parent population,  $P$ , is created by randomly generating  $N$  individuals. Then, mating selection is performed to choose solutions for creating offspring. This process repeats until  $N$  offspring individuals are generated, which forms the offspring population,  $P'$ . Merge  $P$  and  $P'$  to form the combined population  $Q$ . Next, the convergence and the distribution indicators of each individual in  $Q$  are calculated. Finally,

---

#### Algorithm 1 A General Framework of 1by1EA

---

**Require:**  $P$  (population),  $N$  (population size)

- 1: *Initialize*( $P$ )
  - 2: **while** the stopping criterion is not met **do**
  - 3:    $P' = \text{Mating\_selection}(P)$
  - 4:    $P'' = \text{Variation}(P')$
  - 5:    $Q = P \cup P''$
  - 6:   Calculate the convergence indicator  $c(\mathbf{x}_i)$  and distribution indicator  $d(\mathbf{x}_i)$  of each  $\mathbf{x}_i \in Q, i = 1, \dots, N$
  - 7:    $P = \text{One-by-one\_selection}(Q, c, d)$
  - 8: **end while**
  - 9: **return**  $P$
- 

the one-by-one selection procedure is implemented to select  $N$  individuals from  $Q$  to form the population of the next generation.

In the following, we describe in detail the approaches to calculating the convergence and the distribution indicators, the one-by-one selection strategy, and mating selection, which are three important components in 1by1EA.

#### B. Calculation of the Convergence and the Distribution Indicators

1) *The convergence indicator:* In calculating the convergence performance of a solution, each objective is equally emphasized and aggregated into a scalar indicator. A general formulation of calculating the convergence indicator can be summarized as follows:

$$c(\mathbf{x}) = \text{agg}(f_1(\mathbf{x}), \dots, f_M(\mathbf{x})) \quad (2)$$

There are numerous methods for calculating the convergence indicator. In the following, we discuss four metrics, among others, that can measure the convergence performance.

The first one is the sum of all the objectives (Sum for short):

$$c(\mathbf{x}) = \sum_{m=1}^M f_m(\mathbf{x}) \quad (3)$$

Sum is one of the simplest and most well-known aggregation functions to convert multiple objectives into a single one. Intuitively, it is well suited for problems whose Pareto optimal fronts are linear. Additionally, it is usually believed to be unable to solve problems with concave Pareto optimal fronts [42]. However, previous studies [43], [44] have shown that with an adaptive weighted strategy, Sum can also work well for these problems. Another study [45] has shown that it can achieve better search performance than other aggregation approaches in many-objective knapsack problems.

The second one is the Chebyshev distance to the ideal point (CdI for short):

$$c(\mathbf{x}) = \max_{1 \leq m \leq M} |f_m(\mathbf{x}) - z_m^*| \quad (4)$$

where  $\mathbf{z}^* = (z_1^*, \dots, z_M^*)^T$  is the ideal point, and  $z_m^* = \min_{\mathbf{x}_i \in Q} f_m(\mathbf{x}_i)$ . In contrast to Sum, CdI is believed to be capable of finding Pareto solutions on concave fronts [42]. Although CdI is not smooth for continuous objectives, it is able to solve discrete problems, e.g., scheduling problems [46]. However,



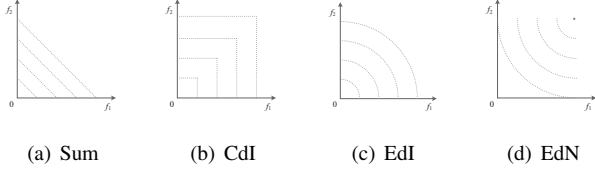


Fig. 2. Contour lines formed by different convergence indicators in the bi-objective space.

CdI is widely used in the decomposition-based MOEAs for solving continuous problems [22], [25], [26], since these methods do not need to compute the derivative of an aggregation function. In addition, CdI is usually termed as the achievement function in the preference-based methods to help the decision maker seek his/her favorite solutions [35], [47], [48].

The third one we consider here is the Euclidean distance to an ideal point (EdI for short):

$$c(\mathbf{x}) = \sqrt{\sum_{m=1}^M (f_m(\mathbf{x}) - z_m^*)^2} \quad (5)$$

Similar to CdI, EdI is advantageous for solving concave problems, especially when their Pareto optimal fronts are a part of hypersphere. It has also been applied in the preference-based methods [39], [49], [50], where the ideal point is usually replaced by a reference point given by the decision maker. In [39], solutions with the smallest values of EdI get the best crowding distances and are preferred in the selection.

The fourth one is the Euclidean distance to the Nadir point (EdN for short):

$$c(\mathbf{x}) = 1 / \sqrt{\sum_{m=1}^M (f_m(\mathbf{x}) - z_m^{\text{nad}})^2} \quad (6)$$

where  $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_M^{\text{nad}})^T$  is the nadir point, and  $z_m^{\text{nad}} = \max_{\mathbf{x}_i \in Q} f_m(\mathbf{x}_i)$ . We employ the reciprocal of the Euclidean distance to the nadir point, since we intend to minimize the convergence indicator. EdN is the invert version of EdI. A recent study [51] has reported that if the distance between the obtained  $\mathbf{z}^*$  and the true ideal point is large, evolving solutions towards  $\mathbf{z}^*$  will guide the population toward a specific region of the Pareto optimal front. In such cases, evaluating solutions based on  $\mathbf{z}^{\text{nad}}$  may be helpful for approximating the entire Pareto optimal front. EdN is often used in decision making [52], [53], where the solution farthest from the nadir point is the best.

All the above four convergence metrics are computationally cheap and easy to obtain. The contour lines formed by different convergence indicators in the bi-objective space are shown in Fig. 2. Solutions lying on the same contour line have the same value of the convergence indicator. As we discussed above, a convergence indicator is advantageous for problems whose Pareto optimal fronts fit its contour lines. Empirical studies of these convergence indicators have been conducted and the results are presented in the supplementary document. It should be noticed that, different from the preference- and decomposition-based methods, the main advantage of our method is that it does not need any weight/reference vector,

which makes it easier to be applied. Since our aim is not to obtain parts of the Pareto optimal front preferred by the decision maker, it is unnecessary to weight the objectives. In the decomposition-based methods, uniformly [23] or adaptively [24], [27] generating the reference vectors is particularly important to achieve good distribution performance. However, in our method, the distribution issue can be resolved by the niche technique based on a distribution indicator.

2) *The distribution indicator*: The distribution indicator is a vector with each of its element indicating the distance between a solution to the rest in the population. Almost all existing distribution performance indicators employ the Euclidean distance to calculate the distance between solutions, or to estimate the solution density. However, as discussed in Section II.B, the Euclidean distance is not well suited for measuring distribution in a high-dimensional space.

The cosine similarity [54] uses the cosine of the angle between two vectors to measure their similarity. Two vectors with exactly the same orientation result in a cosine similarity of 1, and those which are perpendicular with each other produce a cosine similarity of 0. The cosine similarity is within the range of [0,1] regardless of the dimensionality of the vectors. Therefore, the cosine similarity has widely been used for measuring similarity in a high-dimensional space, e.g., in data mining [54], document clustering [55], and intuitionistic fuzzy sets [56]. For many-objective evolutionary optimization, we hypothesize that the cosine similarity will be more effective in distinguishing two solutions in a high-dimensional space than similarity measures based on the Euclidean distance. Fig. 3 provides an illustrative example, where the distribution indicator is the cosine similarity between a solution and the rest, and  $\cos(\theta)$  is used as the threshold to de-emphasize neighboring solutions. Solution A is first selected, and then C is de-emphasized according to the threshold,  $\cos(\theta)$ . Next, solutions B and D are selected, and E is de-emphasized. Finally, F is selected, and I, H, G are de-emphasized. Compared to the selection results illustrated in Fig. 1, in which the Euclidean distance is employed as the distribution indicator, this example shows that utilizing the cosine similarity as the distribution indicator is very helpful in reducing *DRSs* such as solutions G, H, and I.

Therefore, the following distribution indicator is adopted in this study:

$$\mathbf{d}(\mathbf{x}_i) = (d_1(\mathbf{x}_i), \dots, d_{|Q|}(\mathbf{x}_i)), i = 1, \dots, |Q| \quad (7)$$

where

$$d_j(\mathbf{x}_i) = 1 - \cos(\theta_{ij}), j = 1, \dots, |Q| \quad (8)$$

is the distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , and

$$\cos(\theta_{ij}) = \frac{\sum_{m=1}^M (f_m(\mathbf{x}_i) - z_m^*) \cdot (f_m(\mathbf{x}_j) - z_m^*)}{\sqrt{\sum_{m=1}^M (f_m(\mathbf{x}_i) - z_m^*)^2} \cdot \sqrt{\sum_{m=1}^M (f_m(\mathbf{x}_j) - z_m^*)^2}} \quad (9)$$

It is worth noting that  $d_j(\mathbf{x}_i)$  falls in the range of [0, 1], and the smaller  $d_j(\mathbf{x}_i)$ , the closer between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . In addition,  $d_j(\mathbf{x}_i)$  is equal to  $d_i(\mathbf{x}_j)$ , and  $d_i(\mathbf{x}_i) = 0$ . In the selection process, after solution  $\mathbf{x}_i$  is selected, each  $\mathbf{x}_j \in Q$

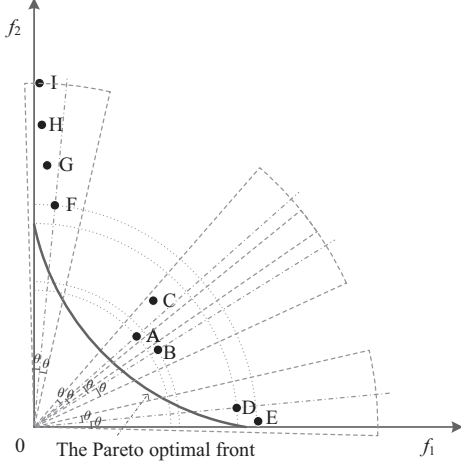


Fig. 3. An example of selecting solutions one by one for a bi-objective problem, where the convergence indicator is the Euclidean distance between a solution and the original point, and the distribution indicator is its cosine similarity to each of the others. Solutions on a quarter circle with dotted lines have the same values of the convergence indicator.  $\cos(\theta)$  is the threshold to de-emphasize neighbors, and the sectors with the dashed lines are the niches formed by  $\cos(\theta)$ . On the basis of the convergence and the distribution indicators, A, B, D, and F are selected in turn, and C, E, and G-I are de-emphasized after A, D, and F are chosen, respectively.

whose distance with solution  $\mathbf{x}_i$ , i.e.,  $d_j(\mathbf{x}_i)$  is smaller than the threshold will be de-emphasized. Using the cosine similarity as the distribution indicator, we can resolve the issue mentioned in Section II.B.

It is interesting to note that the distribution indicator based on the cosine similarity has an **inherent relationship** with the penalty-based boundary intersection (PBI) in MOEA/D [22] and the reference point-based diversity maintenance mechanism in NSGA-III [20]. Taking Fig. 4 as an example, in MOEA/D with the PBI function and NSGA-III, the distribution performance of solution  $\mathbf{x}$  is measured by its distance to the reference vector, i.e.,  $d_2$ , which is calculated using the following expression:

$$d_2 = |\mathbf{x}| \cdot \sqrt{1 - \cos^2 \theta} \quad (10)$$

It can be conceived that  $d_2$  has similar mathematical properties to the distribution indicator based on the cosine similarity. However, our method does not need any predefined reference vectors and is easy to adapt.

### C. The One-by-one Selection Strategy

The one-by-one selection strategy is the key component of lby1EA, and its detailed procedure is described in Algorithms 2, 3, and 4. Based on the convergence and the distribution indicators, individuals in the combined population,  $Q$ , are classified into the following three subsets:  $Q_s$ , the pre-selected individuals (not necessarily be selected),  $Q_{th}$ , the individuals de-emphasized by the distribution threshold, and  $Q_d$ , the individuals dominated by the selected ones.

First, a boundary maintenance mechanism is performed to preserve the corner individuals, **referring to Algorithm 2**, lines

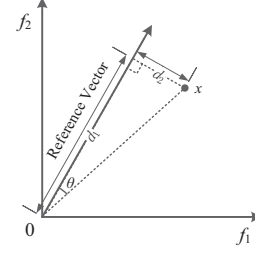


Fig. 4. The distribution performance of solutions in MOEA/D with PBI and NSGA-III.

---

#### Algorithm 2 *One-by-one\_selection*( $Q, c, d$ )

---

**Require:**  $N$  (population size),  $\zeta$  (distribution threshold to de-emphasize individuals)

```

1:  $P = \emptyset$ 
2:  $Q_s = \emptyset$  ▷ set of pre-selected individuals
3:  $Q_{th} = \emptyset$  ▷ set of de-emphasized individuals by  $\zeta$ 
4:  $Q_d = \emptyset$  ▷ set of dominated individuals
5:  $nrank = 1$  ▷ rank used for tournament strategy in mating selection
6: for  $m = 1 \rightarrow M$  do
7:   if  $Q \neq \emptyset$  then
8:     Selecting_the_unique_individual( $c_m(\mathbf{x})$ )
9:     De-emphasizing_individuals( $\mathbf{x}_{\text{unique}}, \zeta$ )
10:     $rank(\mathbf{x}_{\text{unique}}) = nrank$ 
11:   end if
12: end for
13: while  $Q \neq \emptyset$  do
14:   Selecting_the_unique_individual( $c(\mathbf{x})$ )
15:   De-emphasizing_individuals( $\mathbf{x}_{\text{unique}}, \zeta$ )
16:    $rank(\mathbf{x}_{\text{unique}}) = nrank$ 
17: end while
18:  $r = |Q_s|/N$ 
19:  $N_d = |Q_d|$ 
20: if  $|Q_s| \geq N$  then
21:    $P = \{\mathbf{x}_i | \mathbf{x}_i \in Q_s, i \leq N\}$  ▷ add the first  $N$  individuals in  $Q_s$  to  $P$ 
22: else
23:   while  $|Q_s| < N$  do
24:     if  $Q = \emptyset$  then
25:        $Q = Q_{th} \cup Q_d$ 
26:        $Q_{th} = \emptyset$ 
27:        $Q_d = \emptyset$ 
28:        $nrank = nrank + 1$ 
29:     end if
30:     Selecting_the_unique_individual( $c(\mathbf{x})$ )
31:     De-emphasizing_individuals( $\mathbf{x}_{\text{unique}}, \zeta$ )
32:      $rank(\mathbf{x}_{\text{unique}}) = nrank$ 
33:   end while
34:    $P = Q_s$ 
35: end if
36:  $\zeta = \text{update}(\zeta, r, N_d)$ 
37: return  $P$ 

```

---

**Algorithm 3** *Selecting\_the\_unique\_individual*( $f_{agg}(\mathbf{x})$ )

---

```

1:  $\mathbf{x}_{\text{unique}} = \arg \min_{\mathbf{x}_i \in Q} f_{agg}(\mathbf{x}_i)$ 
2:  $Q_s = Q_s \cup \{\mathbf{x}_{\text{unique}}\}$ 
3:  $Q = Q \setminus \{\mathbf{x}_{\text{unique}}\}$ 
4: return  $\mathbf{x}_{\text{unique}}$ 

```

---

**Algorithm 4** *De-emphasizing\_individuals*( $\mathbf{x}_{\text{unique}}, \zeta$ )

---

```

1: for all  $\mathbf{x}_i : d_i(\mathbf{x}_{\text{unique}}) < \zeta, \mathbf{x}_i \in Q$  do
2:    $Q_{th} = Q_{th} \cup \{\mathbf{x}_i\}$ 
3:    $Q = Q \setminus \{\mathbf{x}_i\}$ 
4: end for
5: for all  $\mathbf{x}_i : \mathbf{x}_i \succ \mathbf{x}_{\text{unique}}, \mathbf{x}_i \in Q$  do
6:    $Q_d = Q_d \cup \{\mathbf{x}_i\}$ 
7:    $Q = Q \setminus \{\mathbf{x}_i\}$ 
8: end for
9: return  $Q$ 

```

---

**6-12.** In this study, a corner individual shares a similar concept of the corner solution defined in [57]. If a solution is generated by simultaneously minimizing  $k_c$  ( $k_c < M$ ) objectives of an  $M$ -objective problem, it is referred to as a corner solution. Following the above definition, a corner individual can be defined to be the individual with the smallest value of a scalar function aggregated by  $k_c$  objectives in the current population. Since  $k_c$  can take values from 1 to  $M - 1$ , there are  $2^M - 1$  possible corner solutions lying on the boundaries of the Pareto optimal front. Likewise, there exist  $2^M - 1$  possible corner individuals in a population. As a matter of fact, the number of corner solutions lying on the Pareto optimal front is often much fewer. The  $M$  corner solutions lying on the objective axes have a good ability in representing the bound of each objective on the Pareto optimal front, and can be obtained by simultaneously minimizing  $M - 1$  objectives. Consequently, to achieve the homologous corner individuals, we employ  $M$  aggregated scalar functions that can be termed as variations of the convergence indicator:

$$c_m(\mathbf{x}) = \text{agg}(f_1(\mathbf{x}), \dots, f_{m-1}(\mathbf{x}), f_{m+1}(\mathbf{x}), \dots, f_M(\mathbf{x})), \quad (11)$$

$$m = 1, \dots, M$$

For example,

$$c_m(\mathbf{x}) = \sqrt{\sum_{i=1, i \neq m}^M (f_i(\mathbf{x}) - z_i^*)^2}, m = 1, \dots, M \quad (12)$$

Then the **corner individual**,  $\mathbf{x}_m^{\text{coner}}$ , can be obtained as follows,

$$\mathbf{x}_m^{\text{coner}} = \arg \min_{\mathbf{x}_i \in Q} c_m(\mathbf{x}_i), m = 1, \dots, M \quad (13)$$

**Prioritizing these corner individuals in selection can enhance the algorithm's ability in covering the Pareto optimal front as widely as possible.** Moreover, these individuals are definitely non-dominated individuals [57]. Out of the above reasons, the corner individuals are first selected and put into  $Q_s$ .

Next, the rest individuals in  $Q$  are chosen to put into  $Q_s$  one by one, referring to Algorithm 2, lines 13-17. It should be noted that once an individual is select into  $Q_s$  (lines 8 and

14 in Algorithm 2), the individuals either near this solution or dominated by this solution will be de-emphasized, as shown in Algorithm 2, lines 9 and 15. As a result, no dominated individuals are chosen before any other non-dominated ones, and the problem mentioned in Item (1) in Section II.B can be solved. These selection and de-emphasizing operators will repeat until  $Q$  becomes empty.

When  $Q$  is empty for the first time, if the number of individuals in  $Q_s$  is larger than the predefined population size,  $N$ , only the first  $N$  individuals in  $Q_s$  will be selected into  $P$  (refer to Algorithm 2, lines 20-21); otherwise, the individuals in  $Q_{th}$  and  $Q_d$  will compete to survive until  $N$  individuals are selected into  $Q_s$  (Algorithm 2, lines 23-33). In this process, once  $Q$  is empty, all individuals in  $Q_{th}$  and  $Q_d$  will be moved to  $Q$  (Algorithm 2, lines 24-29).

Last but not the least, the distribution threshold,  $\zeta$ , is updated based on the size of  $Q_s$  (Algorithm 2, line 36) to ensure that  $N$  solutions are selected. The updating method is similar to that used in [21]:

$$\zeta^t = \zeta^{t-1} e^{\frac{1}{M}(\frac{r}{R}-1)} \quad (14)$$

where  $\zeta^{t-1}$  and  $r^{t-1} = |Q_s|/N$  (Algorithm 2, line 18) are the distribution threshold and the ratio of the number of pre-selected individuals to the population size at the  $(t - 1)$ -th generation, respectively. In this study,  $\zeta^1$  is set to 1.  $R \in [0, 2]$  is a threshold to control the number of pre-selected individuals. Intuitively, when  $R = 1$ , the convergence and the distribution performances can be properly balanced. This will be empirically verified in the experiments in the supplementary document. Due to the adaptation in (14), if the number of pre-selected individuals is smaller than  $R \cdot N$ ,  $\zeta^t$  will decrease, and vice versa. Thus, the number of pre-selected individuals remains around  $R \cdot N$ . However, as mentioned in Section II.B, the true Pareto optimal front is unknown during the evolution, implying that it might be misleading to adapt  $\zeta^t$  based on  $r^{t-1}$  and  $\zeta^{t-1}$ . If the number of dominated individuals in  $Q_d$ , i.e.  $N_d$  (Algorithm 2, line 19), is larger than  $(2 - R) \cdot N$  at the  $(t - 1)$ -th generation, the number of pre-selected individuals at the  $(t - 1)$ -th generation cannot be  $N$ , regardless of the value of  $\zeta^{t-1}$ . In such case,  $\zeta^t$  should remain unchanged if  $N_d > (2 - R) \cdot N$ .

#### D. The Mating Selection Strategy

The one-by-one selection strategy described in the previous section is adopted in environmental selection. In the mating selection, a binary tournament selection strategy is proposed to choose promising individuals to form a mating pool. Two tournament strategies are employed, i.e., the ranking method using information from one-by-one selection and the density estimation based on the distribution indicator.

First, two individuals are randomly selected from the parent population. If one has a lower rank than the other, the former is chosen, where the rank,  $\text{rank}(x)$ , can be obtained based on the result of the one-by-one selection at the previous generation. Before  $Q$  becomes empty, the rank of the solutions chosen into  $Q_s$  is set to 1 (Algorithm 2, lines 10 and 16). Once  $Q$  is empty (Algorithm 2, line 24), the rank of the chosen

solutions will be increment by 1 (see Algorithm 2, lines 28 and 32). Note that the rank of each individual at the first generation is set to 1.

Next, if these two individuals have the same rank, we prefer the one with a lower density estimation value,  $d_k(\mathbf{x})$ , which is formulated as follows:

$$d_k(\mathbf{x}) = \frac{1}{\sum_{i=1}^k d_i^{\min}(\mathbf{x}) + 1} \quad (15)$$

where  $d_i^{\min}(\mathbf{x})$ ,  $i = 1, \dots, k$  is one of the  $k$  smallest values in  $\{d_1(\mathbf{x}), \dots, d_{|P|}(\mathbf{x})\}$ . It can be seen that the density estimation method is a variant of the  $k$ -th nearest neighbor method. The difference is that we use the cosine similarity to calculate  $d_j(\mathbf{x})$ ,  $j = 1, \dots, |P|$ , which is the same as in (8), rather than the Euclidean distance.

Finally, if the density estimation fails to distinguish the two individuals, one of them will be chosen randomly.

#### E. Normalization Based on the Corner Solutions

In this study, we further incorporate a normalization procedure into lby1EA based on the corner solutions to enhance its ability in solving scaled problems. This version of lby1EA using corner solution based normalization is termed lby1EA-norm.

Normalization has been demonstrated to be helpful in solving problems whose objective values are disparately scaled [20], [21], [25], [33], [40]. In the normalization, the objective,  $f_m(\mathbf{x})$ ,  $m = 1, 2, \dots, M$ , is replaced by

$$f'_m(\mathbf{x}) = \frac{f_m(\mathbf{x}) - f_m^{\text{lb}}}{f_m^{\text{ub}} - f_m^{\text{lb}}} \quad (16)$$

where  $f_m^{\text{lb}}$  and  $f_m^{\text{ub}}$  should be the lower and the upper bounds of the  $m$ -th objective on the Pareto optimal front, respectively. Since the Pareto optimal front is unknown,  $f_m^{\text{lb}}$  and  $f_m^{\text{ub}}$  are usually estimated based on the obtained solution set.  $f_m^{\text{lb}}$  can be easily set to be the minimal value of  $f_m(\mathbf{x})$  in the current population, whereas the estimation of  $f_m^{\text{ub}}$  requires information about the true Pareto optimal front.

In lby1EA-norm,  $f_m^{\text{ub}}$  is estimated based on  $M$  corner individuals. As mentioned in Section III.C, by simultaneously minimizing  $M - 1$  objectives, a corner solution lying on the objective axes can be obtained, and the value of the objective that is not minimized can represent the upper bound of the objective on the Pareto optimal front. Thus, the homologous corner individuals are well suited for estimating  $f_m^{\text{ub}}$ , and  $f_m^{\text{ub}}$  as follows:

$$f_m^{\text{ub}} = f_m(\mathbf{x}_m^{\text{coner}}), m = 1, \dots, M \quad (17)$$

At each generation in lby1EA-norm, before calculating the convergence and the distribution indicators, each individual in the combined population,  $Q$ , will be normalized using (16). It should be noted that after normalization, the ideal point,  $\mathbf{z}^*$ , used for calculating the convergence and the distribution indicators should be set to  $(0, \dots, 0)$ , whereas the value of each element of the nadir point,  $\mathbf{z}^{\text{nad}}$ , may be larger than 1. Owing to this normalization procedure, lby1EA-norm is particularly capable of solving scaled problems, which will be empirically shown in the experiments.

TABLE I  
THE CHARACTERISTICS OF THE TEST PROBLEMS, WHERE  $k_w$  IS THE POSITION-RELATED PARAMETER WHOSE VALUE IS  $M - 1$ .

Problem	$n$	Characteristics
DTLZ1	$M+4$	Linear, Multimodal
DTLZ2	$M+9$	Concave
DTLZ3	$M+9$	Concave, Multimodal
DTLZ4	$M+9$	Concave, Biased
DTLZ5	$M+9$	Concave, Degenerate
DTLZ6	$M+9$	Concave, Degenerate, Biased
DTLZ7	$M+19$	Mixed, Disconnected, Multimodal, Scaled
WFG1	$k_w+24$	Mixed, Biased, Scaled
WFG2	$k_w+24$	Convex, Disconnected, Multi-modal, Non-separable, Scaled
WFG3	$k_w+24$	Linear, Degenerate, Non-separable, Scaled
WFG4	$k_w+24$	Concave, Multimodal, Scaled
WFG5	$k_w+24$	Concave, Biased, Scaled
WFG6	$k_w+24$	Concave, Non-separable, Scaled
WFG7	$k_w+24$	Concave, Biased, Scaled
WFG8	$k_w+24$	Concave, Biased, Non-separable, Scaled
WFG9	$k_w+24$	Concave, Biased, Multi-modal, Deceptive, Non-separable, Scaled

#### IV. EXPERIMENTAL DESIGN

This section describes the experimental design for examining the performance of lby1EA. The test problems and the performance indicators used in the experiments are given at first. Then, eight state-of-the-art MOEAs to be compared, i.e., KnEA [21], BiGE [40], EFR-RR [25], MOEA/D-ACD [26], MOMBI-II [33], NSGA-III [20], GrEA [29], and SDE [19], are briefly introduced. Finally, the commonly used parameters are set for the comparative studies of these algorithms.

##### A. Test Problems and Performance Indicators

DTLZ [58] and WFG [59], two widely used test suites, are adopted for empirical comparisons in this study. These test suites are composed of optimization problems having linear, concave, multimodal, disconnected, biased, scaled, or degenerate Pareto optimal fronts. The characteristics of the Pareto optimal front of different test problems are summarized in Table I. In this paper, we consider these problems with 3, 6, 8, 10 and 15 objectives. A detailed description of the DTLZ and WFG suites can be found in [58] and [59].

In order to measure the performance of different algorithms on these test problems, the inverted generation distance plus (IGD<sup>+</sup>) [60] is adopted in this work. IGD<sup>+</sup> is a modified version of IGD [61], which is a widely used performance indicator. IGD can measure both the convergence and diversity performance of a solution set obtained by an algorithm, and the smaller the value of IGD, the better the performance of the algorithm. However, IGD is a Pareto non-compliant indicator, thus it may result in inaccurate evaluation in some cases. In IGD<sup>+</sup>, the distance between a solution and a reference point is refined by taking their Pareto dominance relationship into account, which makes it weakly Pareto compliant and more accurate on evaluation. Similar to IGD, IGD<sup>+</sup> requires a reference set and the reference points are typically uniformly distributed on the Pareto optimal front of a test problem. In this work, we set the number of reference points to around 5,000 and 10,000 when  $M \in \{3, 6, 8, 10\}$  and  $M = 15$ ,



respectively. It should be noticed that when calculating  $IGD^+$  for scaled problems, both solutions and reference points will be normalized based on the true Pareto optimal front.

### B. Compared Algorithms

The following eight state-of-the-art MOEAs are chosen for comparison to assess the performance of the proposed algorithm: KnEA [21], BiGE [40], EFR-RR [25], MOEA/D-ACD [26], MOMBI-II [33], NSGA-III [20], GrEA [29], and SDE [19]. These algorithms cover all the main categories mentioned in Section I for many-objective optimization. The code of these algorithms except for NSGA-III are all from the authors. NSGA-III is implemented by the authors of KnEA.

- KnEA [21] was established under the framework of NSGA-II [10], and it gives a priority to the knee points among non-dominated solutions. The process of seeking local knee points can also be seen as a one-by-one selection procedure, in which the distance between a solution and the hyperplane defined by the extreme solutions is used as the convergence indicator, and the Chebyshev distance between solutions is utilized for the distribution indicator. Hence, KnEA is very relevant to the proposed algorithm.
- BiGE [40] also simultaneously considers the convergence and distribution performance of a solution. Different from lby1EA, BiGE transforms an MaOP into a bi-objective problem whose objectives are the convergence and distribution indicators. A solution set with well-balanced performance can be achieved by optimizing the bi-objective optimization problem. However, the niche technique used in BiGE is based on the Euclidean distance and the niche size cannot be tuned adaptively.
- EFR-RR [25] is based on the framework of NSGA-II. However, it is inherently a decomposition-based method like MOEA/D [22]. EFR-RR considers the distribution performance of a solution firstly. Only when the perpendicular distance from the solution to the reference vector is small, its scalarizing function value will be calculated. This method suggests another idea on balancing convergence and diversity in the high-dimensional objective space and has also been applied to MOEA/D, termed MOEA/D-DU. Since EFR-RR performs slightly better than MOEA/D-DU according to the original study, we choose the former for comparison.
- MOEA/D-ACD [26] was developed from the original MOEA/D. The goal of this method is similar to EFR-RR. However, it achieves this goal by adding constraints to the scalarizing functions. In this way, the angle between a solution and the reference point is limited by a threshold, which guarantees the distribution performance of the solution. Moreover, this method offers a strategy for adaptively adjusting the threshold.
- MOMBI-II [33] is an indicator-based algorithm that uses the R2 indicator to guide the search. The R2 indicator is an attractive alternative to hypervolume [32], due to its low computational cost and weak-Pareto compatibility. MOMBI-II takes two key aspects into account, i.e., using

the scalarizing function and statistical information about the population's proximity to the true Pareto optimal front.

- NSGA-III [20] was adapted from a popular dominance-based MOEA, NSGA-II, for handling MaOPs. NSGA-III uses a reference-point-based selection criterion instead of a density-based counterpart (crowding distance) in NSGA-II. A two-layer strategy was proposed for generating well-distributed reference points in the high-dimensional objective space, which can also be used in MOEA/D.
- GrEA [29] exploits the potential of the grid-based approach to deal with MaOPs. Grid dominance and grid difference are introduced to determine the relationship between individuals in a grid environment. Three grid-based criteria are incorporated into the fitness of an individual to distinguish individuals in both mating and environmental selection processes. Moreover, a fitness adjustment strategy was developed to avoid partial overcrowding as well as to guide the search towards various directions in the archive.
- SDE [19] is a recently proposed method that modifies the density estimation strategies in traditional Pareto-based MOEAs. By shifting the position of an individual in a sparse region, individuals with poor convergence can be assigned with a high density value, and then discarded during the selection. In this study, the version that integrates SDE into SPEA2 [11] (denoted as SPEA2+SDE) is employed due to its better performance compared to other variants.

### C. Parameter Settings

The following parameter settings are adopted by all compared algorithms. Simulated binary crossover and polynomial mutation are used as the crossover and mutation operators, with both distribution indexes being set to 20. The crossover and mutation probabilities are 1.0 and  $1/n$ , respectively, where  $n$  is the number of decision variables. These settings of the evolutionary operators are the same as those in the recent studies and have been demonstrated to be able to generate good solutions [19], [21], [25], [29], [40]. The termination criterion is the predefined maximum number of generations. For difficult problems, i.e., DTLZ1, DTLZ3, DTLZ6 and WFG1, the maximum number of generations is set to 1,000 so that the algorithms are able to converge sufficiently to the true Pareto optimal front. For the other test problems, the maximum number of generations is set to 300. To avoid the situation in which all reference points locate on the boundary of the Pareto optimal front for problems with a large number of objectives, the strategy of two-layered reference points is used for EFR-RR, MOEA/D-ACD, MOMBI-II and NSGA-III. As a result, the population size of EFR-RR, MOEA/D-ACD and NSGA-III cannot be arbitrarily specified. For a fair comparison, we set the population size of the other algorithms under comparison to the same value as these three algorithms. The setting of the population size,  $N$ , and the parameters for controlling the number of reference points are listed in Table II.

TABLE II

SETTING OF THE POPULATION SIZE, WHERE  $p_1$  AND  $p_2$  ARE PARAMETERS CONTROLLING THE NUMBER OF REFERENCE POINTS ALONG THE BOUNDARY OF THE PARETO OPTIMAL FRONT AND INSIDE IT, RESPECTIVELY.

$M$	$p_1$	$p_2$	$N$
3	13	0	105
6	4	1	132
8	3	2	156
10	3	2	275
15	2	1	135

TABLE III

THE PARAMETER SETTINGS IN KNEA AND GREa, WHERE THE VALUES OF BOTH  $T$  AND  $div$  CORRESPOND TO THE NUMBER OF OBJECTIVES OF A PROBLEM.

$M$	$T$ in KnEA					$div$ in GrEA				
	3	6	8	10	15	3	6	8	10	15
DTLZ1	0.5	0.2	0.1	0.1	0.1	11	11	11	13	28
DTLZ2	0.5	0.5	0.5	0.5	0.5	11	8	8	9	10
DTLZ3	0.5	0.2	0.1	0.1	0.1	12	15	15	15	28
DTLZ4	0.5	0.5	0.5	0.5	0.5	11	8	8	9	10
DTLZ5	0.5	0.5	0.3	0.3	0.2	38	16	11	11	12
DTLZ6	0.5	0.4	0.3	0.3	0.2	40	50	50	50	50
DTLZ7	0.5	0.5	0.5	0.4	0.4	10	8	6	4	4
WFG1	0.5	0.5	0.5	0.5	0.5	5	6	8	10	15
WFG2	0.5	0.5	0.5	0.5	0.5	10	9	9	9	17
WFG3	0.5	0.5	0.5	0.5	0.5	18	18	18	24	22
WFG4&9	0.5	0.5	0.3	0.3	0.3	10	11	11	14	12
WFG5-8	0.5	0.5	0.5	0.5	0.5	10	11	11	14	12

In 1by1EA,  $k$  is set to  $0.1N$  to balance the computational cost and the accuracy in estimating the density.  $R$  is set to 1 to balance convergence and distribution as discussed in Section III.C. The Sum method is adopted as the convergence indicator for DTLZ1 and WFG3, since they have linear Pareto optimal fronts. The Pareto optimal fronts of DTLZ7, WFG1 and WFG2 are complex, and can be roughly regarded as different hyperplanes. As a result, the Sum method is also applied to these problems. Additionally, EdI is applied to the other problems of DTLZ and WFG, since all these problems are concave. As suggested in the original studies, the Chebychev approach and the PBI approach with  $\theta = 5$  are chosen as the scalarizing method in EFR-RR and MOEA/D-ACD, respectively. The neighborhood size is set to  $0.1N$  in both algorithms. Additionally,  $K$  is set to 2 in EFR-RR. Since both the number of objectives and the population size are different from those of the original KnEA and GrEA, we adjust the threshold,  $T$ , in KnEA and the grid division,  $div$ , in GrEA according to the guidelines provided in the original studies to achieve the best performances of these algorithms. The settings of  $T$  and  $div$  for DTLZ and WFG problems are listed in Table III.

Each algorithm is run for 20 times on each optimization problem, and the mean value of  $IGD^+$  are calculated. In addition, the Wilcoxon's rank sum test is employed to determine whether one algorithm has a statistically significant difference with the other on the performance indicator, and the null hypothesis is rejected at a significant level of 0.05.

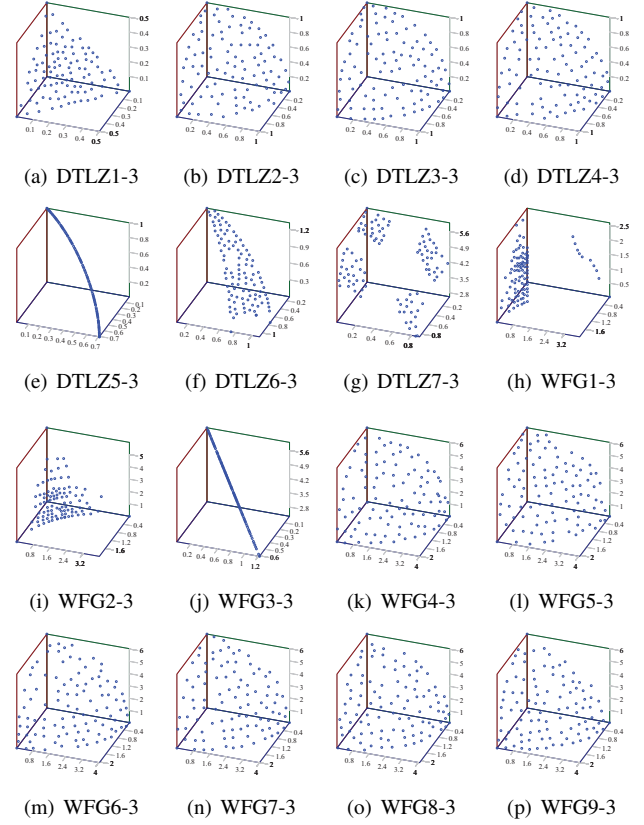


Fig. 5. Pareto fronts achieved by 1by1EA on 3-objective DTLZ and WFG problems.

## V. RESULTS AND DISCUSSIONS

In this section, the performances of 1by1EA are empirically evaluated. The experiments are divided into the following two parts. The first part investigates the achieved Pareto fronts of 1by1EA on 3-objective problems, and the second compares 1by1EA with the other eight state-of-the-art MOEAs.

### A. Pareto fronts achieved by 1by1EA on 3-objective DTLZ and WFG problems

Before comparing the proposed 1by1EA with the other algorithms on MaOPs, we visualize the Pareto fronts achieved by 1by1EA on 3-objective DTLZ and WFG problems for better a understanding of its performance. Since DTLZ7 and WFG1-9 are scaled problems, 1by1EA-norm was adopted, whereas the original 1by1EA was adopted on DTLZ1-6. Note that the population size was set to 100 in this subsection. Fig. 5 shows the 3D plots of the final solution sets of a single run in the objective space. This particular run is associated with the result which is the closest to the mean  $IGD^+$  value. Note that in this paper, DTLZA-B refers to DTLZA with  $B$  objectives, and WFGA-B has a similar meaning.

It is evident from Fig. 5 that the solutions are very close to the Pareto optimal fronts, and they are distributed in the whole region of the fronts for most problems. Although 1by1EA achieves good performance on some degenerate problems, e.g., DTLZ5 and WFG3, it struggles to converge on DTLZ6 because of bias in its objectives. For the disconnected problems,

e.g., DTLZ7 and WFG1, 1by1EA can obtain all parts of their Pareto front and the solutions are distributed very well on DTLZ7.

### B. Comparison with State-of-the-art Algorithms

In this subsection, 1by1EA and 1by1EA-norm are compared with KnEA, BiGE, EFR-RR, MOEA/D-ACD, MOMBI-II, NSGA-III, GrEA, and SDE on DTLZ and WFG with no less than four objectives. We divide these problems into the following two groups. The first group is the normalized problems, i.e., DTLZ1 to 6, and the second is the scaled problems, i.e., DTLZ7 and WFG1 to 9. Tables IV and VI show the ranks of different algorithms in terms of their  $IGD^+$  values on the normalized problems and the scaled problems, respectively, where the results with higher ranks are shaded more brightly. ‘†’ and ‘‡’ indicate that the result is significantly different from that of 1by1EA and 1by1EA-norm, respectively, and ‘\*’ suggests that the result is significantly different from both of them. At the bottom of Tables IV and VI, the average rank (denoted by Avg.Rk) of each algorithm are also given, where the significance test are based on the ranks on each instance. Tables V and VII list the summary of the significance tests on  $IGD^+$  between 1by1EA (1by1EA-norm) and the other algorithms under comparison. When  $A$  vs.  $B$ , ‘+’ (‘-’) indicates the number of instances on which the results of  $A$  are significantly better (worse) than those of  $B$ , and ‘=’ means the number of instances where there exists no statistical significance between the results of  $A$  and  $B$ . The detailed values of  $IGD^+$  are provided in the supplementary document.

TABLE V  
SUMMARY OF THE SIGNIFICANCE TEST OF  $IGD^+$  BETWEEN THE PROPOSED ALGORITHMS AND THE OTHER ALGORITHMS ON DTLZ1 TO 6.

	1by1EA vs.			1by1EA-norm vs.		
	+	-	=	+	-	=
1by1EA	\	\	\	3	18	9
1by1EA-norm	18	3	9	\	\	\
KnEA	20	7	3	17	11	2
BiGE	25	2	3	20	5	5
EFR-RR	18	9	3	15	14	1
MOEA/D-ACD	22	6	2	18	10	2
MOMBI-II	15	10	5	8	16	6
NSGA-III	18	8	2	13	12	5
GrEA	26	1	3	22	6	2
SPEA2+SDE	11	8	11	6	20	4

From Tables IV- VII we can draw the following conclusions. Generally, 1by1EA and 1by1EA-norm perform best on the normalized and scaled problems, respectively, among the ten compared algorithms, according to their average ranks. Although the statistical analysis in terms of rank does not show that 1by1EA (1by1EA-norm) has a significant better performance than the second and the third best algorithms, we can see from Table V ( VII) that 1by1EA (1by1EA-norm) significantly outperforms them on most test instances. Since the original 1by1EA lacks the normalization procedure, it does not work as well as 1by1EA-norm on the scaled problems, demonstrating the effectiveness of the normalization procedure for scaled problems. By contrast, 1by1EA-norm does not behave as well as 1by1EA on the normalized

TABLE VII  
SUMMARY OF THE SIGNIFICANCE TEST OF  $IGD^+$  BETWEEN THE PROPOSED ALGORITHMS AND THE OTHER ALGORITHMS ON DTLZ7 AND WFG1 TO 9.

	1by1EA vs.			1by1EA-norm vs.		
	+	-	=	+	-	=
1by1EA	\	\	\	42	7	1
1by1EA-norm	7	42	1	\	\	\
KnEA	9	39	2	27	15	8
BiGE	14	36	0	24	16	10
EFR-RR	9	39	2	26	16	8
MOEA/D-ACD	47	1	2	50	0	0
MOMBI-II	20	25	5	42	6	2
NSGA-III	12	35	3	25	16	9
GrEA	7	36	7	26	14	10
SPEA2+SDE	7	37	6	33	11	7

problems, because the normalization procedure may transform the original objectives into wrong scales when solving a normalized problem, especially if the problem has a huge number of local PFs, e.g., DTLZ1 and DTLZ3. It is interesting to note that 1by1EA (1by1EA-norm) shows a clearly better performance on DTLZ5 (WFG3). DTLZ5 (WFG3) has  $M - 1$  redundant objectives, and its Pareto optimal front is a degenerate curve (line). Previous studies [62] have shown that the corner solutions are effective for dimensionality reduction. In 1by1EA (1by1EA-norm), solutions are located within the boundaries formed by the corner solutions, which has the same effect as dimensionality reduction. That is why 1by1EA (1by1EA-norm) outperforms the others on DTLZ5 (WFG3). It is worth mentioning that although DTLZ5 and DTLZ6 have the same Pareto optimal front, 1by1EA cannot produce satisfactory results on the latter, since the bias in DTLZ6 objectives results in wrong corner solutions. This phenomenon can be observed visually in Fig. 5(f). It is also interesting to compare the other results on 3-objectives problems in Fig. 5 with those in Tables IV and VI, where 1by1EA (1by1EA-norm) achieves relatively good performance on DTLZ2, 3, 4, 5 and WFG3, 4, 5, 8, 9.

KnEA has an advantage of approximating the Pareto optimal front by giving preferences to the knee points among non-dominated solutions. It has a normalization procedure based on the minimal and the maximal objectives of the current population. This normalization procedure makes KnEA competitive on the scaled problems. However, it also makes KnEA worse than 1by1EA-norm on DTLZ1 to 6.

BiGE achieves poor  $IGD^+$  values on the problems that have a large range of objectives, e.g., DTLZ1 and DTLZ3. The main reason is that BiGE uses a niche technique based on the Euclidean distance, and the niche size is not adjustable. However, because of its normalization procedure, it works well on most scaled problems, especially WFG2.

EFR-RR can effectively deal with both the normalized and the scaled problems except for DTLZ5, DTLZ6 and WFG3. This attributes to the fact that the reference vectors employed in EFR-RR are uniformly distributed in the whole objective space, whereas the Pareto optimal solutions of these problems are not. Thus, the performance of EFR-RR can be further improved by integrating a strategy for adaptively generating

TABLE IV  
RANKS OF DIFFERENT ALGORITHMS ON DTLZ1 TO 6 IN TERMS OF MEAN VALUES OF IGD<sup>+</sup>.

Objective	1by1EA					1by1EA-norm					KnEA					BiGE					EFR-RR					MOEA/D-ACD					MOMBI-II					NSGA-III					GrEA					SPEA2+SDE					
	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15											
DTLZ1	5‡	2‡	5‡	5‡	4‡	9†	7†	7†	6†	6†	8*	8*	9*	8*	8*	10*	10*	10*	10*	9*	4	4*	1*	1*	2*	6*	6*	3*	2*	3*	2	5*	6†	7†	7†	1*	3	3‡	2*	4*	5*	7*	9*	8*	9*	10*	3‡	1‡	4*	3*	1*
DTLZ2	2	4	4	7‡	6‡	5	1	3	3†	1†	9*	7*	8*	8*	3*	10*	9*	9*	5*	4‡	4*	8*	6*	4‡	2*	7*	10*	10*	10*	10*	3*	3	1	1†	9*	1	2	2†	2†	7*	8*	6†	5‡	7*	9*	7*	8*	6†	5‡	6‡	5‡
DTLZ3	3‡	2‡	1‡	3‡	1	6†	5†	5†	5†	3	7*	7*	8*	8*	10*	9*	10*	10*	10*	8*	5*	9*	4*	1‡	4*	10*	8*	6*	6*	6*	1	1‡	2‡	4†	5*	2	3*	7†	7*	8*	6†	9*	9*	9*	9*	4*	4‡	3‡	2‡	2	
DTLZ4	4‡	3‡	3‡	4‡	7	7	9†	9†	7†	5	3‡	4‡	6*	5‡	3‡	6*	6†	5†	10*	4*	1*	1*	1*	2*	1*	2*	6	10*	9*	10*	5*	5*	5*	2‡	1*	2*	9*	10*	7*	3*	8*	10*	2‡	7*	8†	9*	8*	8*	4‡	6*	6
DTLZ5	1‡	1‡	1‡	1‡	1‡	6†	8†	7†	4†	3†	8†	4†	7†	5*	5*	10†	4†	6*	6*	6*	7*	10*	7*	7*	4*	9*	2	2*	2*	2*	8*	6*	9*	10*	9*	5*	9*	10*	9*	7*	3*	5*	5*	8*	9*	10*	2*	3*	3*	4†	3*
DTLZ6	7	6	4	6	6‡	8	7	5	5	1†	1*	3*	2*	3*	4*	6*	5	6*	7*	7*	10*	10*	10*	10*	9*	9*	8*	7*	1*	3*	3*	2*	3*	4*	5*	2*	4*	8*	8*	8*	4*	9*	9*	9*	10*	5*	1*	1*	2*	2*	
Avg.Rk	3.63‡					5.46†					5.86†					7.56*					4.96†					6.16†					4.36‡					5.43†					7.63*					3.83‡					

TABLE VI  
RANKS OF DIFFERENT ALGORITHMS ON DTLZ7 AND WFG1 TO 9 IN TERMS OF MEAN VALUES OF IGD<sup>+</sup>.

Objective	1by1EA					1by1EA-norm					KnEA					BiGE					EFR-RR					MOEA/D-ACD					MOMBI-II					NSGA-III					GrEA					SPEA2+SDE				
	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15	3	6	8	10	15										
DTLZ7	4*	4*	2*	4*	4*	8*	5*	7*	7*	1*	2*	2*	4*	6*	9*	6*	8*	8*	8*	6*	7*	9*	3*	2*	2*	10*	10*	10*	10*	10*	9*	7*	9*	9*	5*	5*	6*	6*	5*	3*	3*	1*	5*	3*	7*	1*	3*	1*	1*	8*
WFG1	8*	5*	4*	3*	3*	7*	6*	6*	4*	2*	3*	4*	5*	7*	9*	9*	9*	9*	8*	8*	4*	7*	7*	6*	1*	10*	10*	10*	10*	10*	1*	1*	1*	1*	4*	5*	8*	8*	9*	7*	6*	2*	2*	5*	6*	2*	3*	3*	2*	5*
WFG2	9*	9*	9*	8*	5*	5*	6*	6*	5*	2*	2*	3*	2*	7*	7*	3*	1*	1*	1*	1*	8*	5*	3*	2*	6*	10*	10*	10*	10*	9*	7*	7*	7*	4*	10*	6*	8*	8*	9*	3*	4*	2*	4*	6*	8*	1*	4*	5*	3*	4*
WFG3	9*	3*	3*	2*	2*	1*	1*	1*	1*	1*	7*	5*	7*	7*	9*	5*	4*	4*	4*	4*	6*	8*	5*	5*	5*	10*	10*	10*	10*	10*	2*	7*	9*	9*	7*	8*	9*	6*	6*	4*	4*	2*	2*	3*	3*	3*	6*	8*	8*	8*
WFG4	9*	8*	9*	9*	7*	1*	1*	4*	5*	5*	7*	6*	6*	4*	2*	4*	4*	2*	2*	3*	3*	2*	1*	1*	4*	10*	10*	10*	10*	9*	5*	9*	8*	8*	10*	6*	3*	3*	3*	1*	8*	5*	5*	7*	8*	2*	7*	7*	6*	6*
WFG5	9*	8*	9*	9*	6*	3*	3*	6*	6*	5*	6*	6*	4*	5*	2*	5*	5*	5*	3*	4*	8*	4*	2*	2*	3*	10*	10*	10*	10*	9*	7*	9*	8*	8*	10*	1*	1*	1*	1*	4*	2*	3*	4*	7*	2*	7*	7*	7*	8*	
WFG6	9*	8*	9*	9*	8*	5*	1*	3*	6*	5*	8*	6*	6*	4*	2*	6*	4*	4*	2*	4*	7*	5*	2*	3*	1*	10*	10*	10*	10*	9*	3*	9*	8*	7*	10*	4*	2*	1*	1*	3*	2*	3*	5*	5*	6*	1*	7*	7*	8*	7*
WFG7	9*	9*	9*	9*	8*	5*	2*	5*	6*	5*	6*	6*	6*	5*	2*	8*	5*	4*	3*	3*	7*	4*	1*	2*	4*	10*	10*	10*	10*	9*	4*	8*	8*	8*	10*	3*	3*	2*	1*	1*	2*	1*	3*	4*	7*	1*	7*	7*	6*	6*
WFG8	9*	8*	9*	9*	8*	2*	1*	1*	1*	1*	8*	7*	5*	3*	3*	7*	5*	2*	2*	5*	5*	4*	4*	4*	6*	10*	10*	10*	10*	9*	6*	9*	8*	7*	10*	4*	2*	3*	6*	2*	3*	3*	6*	8*	7*	1*	6*	7*	5*	4*
WFG9	9*	8*	9*	9*	6*	1*	1*	3*	4*	5*	5*	4*	6*	5*	4*	2*	2*	2*	1*	3*	6*	5*	1*	2*	2*	10*	10*	10*	10*	9*	7*	9*	8*	8*	10*	8*	6*	5*	3*	1*	3*	3*	4*	6*	8*	4*	6*	7*	7*	7*
Avg.Rk	7.08*					3.68*					5.12*					4.36*					4.12*					9.86*					7.10*					4.26*					4.40*					5.00*				

the reference vectors [24], [27].

MOEA/D-ACD achieves appealing results on normalized problems. We notice that the number of non-dominated solutions in the final set is fewer than the population size in most cases, which makes MOEA/D-ACD less advantageous. In addition, MOEA/D-ACD performs poorly on all scaled problems. This does not mean that MOEA/D-ACD is ineffective for dealing with MaOPs. The main reason is that it lacks a normalization procedure, which confirms that normalization is particularly important in solving scaled problems. However, 1by1EA without normalization so significantly performs better than MOEA/D-ACD on scaled problems.

Although MOMBI-II has a normalization procedure, it performs relatively well on most normalized problems. This can be attributed to its advanced strategy for updating the nadir point, which reduces the risk of transforming the original objectives to wrong scales. However, MOMBI-II does not achieve satisfactory results on the scaled problems except for WFG1.

NSGA-III shows an overall competitive performance on both the normalized and scaled problems except for DTLZ5, DTLZ6 and WFG3 due to the same weakness of EFR-RR. Moreover, since DTLZ4 has a non-uniform Pareto optimal front, NSGA-III cannot maintain a good diversity like EFR-RR.

GrEA achieves encouraging results on most scaled problems and some low-dimensional normalized problems. By dividing each dimension of the objective space into the same number of divisions, GrEA has an inherent ability of normalization. The main drawback of GrEA is that its performance is very sensitive to the value of parameter *div*. In our experiments, we have tested a number of settings of *div* for each instance to make sure that GrEA can obtain a satisfactory performance. If a method of adaptively adjusting *div* can be further developed,

the performance of GrEA will be considerably enhanced on various problems.

SPEA2+SDE works quite well on most normalized problems, and generally has the medium performance on the scaled problems among the compared algorithms. However, it works relatively poorly for the scaled problems with more than 3 objectives. We speculate that the bias in the objectives and the high-dimensional objective space make the shifting procedure less efficient in estimating the density of solutions. This remains to be examined in future work.

As a whole, 1by1EA and 1by1EA-norm outperform the compared algorithms on the normalized and scaled problems, respectively. It is worth noting that the normalization procedure has a great influence on most existing algorithms. Having incorporated the normalization procedure, 1by1EA-norm, KnEA, BiGE, EFR-RR, NSGA-III, and GrEA perform well on scaled problems, but relatively poorly on normalized problems. On the other hand, 1by1EA, MOEA/D-ACD and SPEA2+SDE behave in an opposite way. The normalization procedure in MOMBI-II is inspiring, but needs further investigation. Hence, when and how to apply the normalization procedure is an important issue for future studies in many-objective optimization. Notwithstanding this, for the normalized problems, 1by1EA-norm performs consistently better compared to KnEA, BiGE, MOEA/D-ACD and GrEA. For the scaled problems, the original 1by1EA performs better than MOEA/D-ACD. Moreover, unlike KnEA and GrEA, 1by1EA (1by1EA-norm) is relatively insensitive to its parameters and does not require the generation of reference vectors as required in EFR-RR, MOEA/D-ACD, MOMBI-II and NSGA-III. Therefore, we can conclude that 1by1EA (1by1EA-norm) is very competitive compared with the state-of-the-art of many-objective evolutionary algorithms.



## VI. CONCLUSION

In this paper, we have proposed a novel many-objective evolutionary algorithm, termed lbylEA, in which solutions are selected one by one. In lbylEA, solutions are selected according to the convergence and the distribution indicators. The former measures the distance between a solution and the Pareto optimal front, while the latter measures its distance to each other. The environmental selection strategy of lbylEA consists of the following two important steps. In the first step, only one solution with the best value of the convergence indicator is selected, focusing on the convergence. In the second step, solutions close to the one selected in the first step are de-emphasized according to the distribution indicator, thus maintaining the diversity of the population. By repeating the above two steps, a solution set with both good convergence and distribution performances can be obtained.

To demonstrate the effectiveness of lbylEA, we test its performance on DTLZ, WFG and Pareto-Box (refer to the supplementary document) problems in comparison with eight state-of-the-art algorithms, namely, KnEA, BiGE, EFR-RR, MOEA/D-ACD, MOMBI-II, NSGA-III, GrEA, and SDE. The experimental results demonstrate that both lbylEA and its variant with normalization are very competitive among the compared algorithms on most problems studied in this work.

In this study, several methods for calculating the convergence indicators are considered. In the supplementary document, we empirically show that a convergence indicator can improve the performance of lbylEA when the contour lines formed by the convergence indicator have a similar shape to that of the Pareto optimal front of an optimization problem. However, the shape of the Pareto optimal front of a practical optimization problem is frequently unknown beforehand. A similar issue of selecting appropriate scalarizing functions for MOEA/D has been recently reported in [63], which is solved by adaptively approximating the optimal parameter value in the scalarizing functions. Therefore, our future work includes a mechanism that can adaptively choose an appropriate convergence indicator or use an ensemble of multiple convergence indicators during the evolution of lbylEA.

In addition, it has been found that the normalization procedure based on the particular corner solutions can further enhance lbylEA's capability in handling scaled problems. However, the normalization procedure may also convert an objective into a wrong scale when solving some complex problems. To promote the precision and the reliability of the normalization procedure, estimating the bound of an objective based on statistical information about the corner solutions during the evolution of lbylEA is of great potential.

## ACKNOWLEDGMENT

This work was jointly supported by National Natural Science Foundation of China with grant No. 61375067 and 61403155, National Basic Research Program of China (973 Program) with grant No. 2014CB046306-2, Innovation Project for College Graduates of Jiangsu Province with grant No. KYLX15\_1422, and Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of National Natural Science Foundation of China with grant No. 61428302.

## REFERENCES

- [1] Y. Han, D. Gong, Y. Jin, and Q.-k. Pan, "Evolutionary multi-objective blocking lot-streaming flow shop scheduling with interval processing time," *Applied Soft Computing*, vol. 42, pp. 229–245, 2016.
- [2] Y. Li, S. Sui, and S. Tong, "Adaptive fuzzy control design for stochastic nonlinear switched systems with arbitrary switchings and unmodeled dynamics," *Cybernetics, IEEE Transactions on*, 2016.
- [3] Y. Li and S. Tong, "Adaptive fuzzy output-feedback stabilization control for a class of switched nonstrict-feedback nonlinear systems," *Cybernetics, IEEE Transactions on*, 2016.
- [4] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Information sciences*, vol. 192, pp. 213–227, 2012.
- [5] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *Computational Intelligence Magazine, IEEE*, vol. 4, no. 3, pp. 62–76, 2009.
- [6] P. M. Reed and D. Hadka, "Evolving many-objective water management to exploit exascale computing," *Water Resources Research*, vol. 50, no. 10, pp. 8367–8373, 2014.
- [7] M. Li, S. Yang, and X. Liu, "Diversity comparison of Pareto front approximations in many-objective optimization," *Cybernetics, IEEE Transactions on*, vol. 44, no. 12, pp. 2568–2584, 2014.
- [8] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 13, 2015.
- [9] P. C. Roy, M. Islam, K. Murase, X. Yao *et al.*, "Evolutionary path control strategy for solving many-objective optimization problem," *Cybernetics, IEEE Transactions on*, vol. 45, no. 4, pp. 702–715, 2015.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationssnetze (TIK), Tech. Rep., 2001.
- [12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 770–784, 2007.
- [13] K.-i. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2. IEEE, 2001, pp. 957–962.
- [14] H. Sato, H. Aguirre, and K. Tanaka, "Improved s-cdas using crossover controlling the number of crossed genes for many-objective optimization," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 753–760.
- [15] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 5, pp. 1402–1412, 2008.
- [16] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 269–285, 2014.
- [17] A. López Jaimes and C. A. Coello Coello, "Study of preference relations in many-objective optimization," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 2009, pp. 611–618.
- [18] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 2, pp. 183–195, 2011.
- [19] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 348–365, 2014.
- [20] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 577–601, 2013.
- [21] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.
- [22] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [23] Y.-y. Tan, Y.-c. Jiao, H. Li, and X.-k. Wang, "MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with



- many objectives,” *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.
- [24] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, “MOEA/D with adaptive weight adjustment,” *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
- [25] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, “Balancing convergence and diversity in decomposition-based many-objective optimizers,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 180–198, 2016.
- [26] L. Wang and Q. Zhang, “Constrained subproblems in decomposition based multiobjective evolutionary algorithm,” *IEEE Transactions on Evolutionary Computation*, 2015.
- [27] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, “A reference vector guided evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, 2016.
- [28] K. Deb, M. Mohan, and S. Mishra, “Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions,” *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.
- [29] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 5, pp. 721–736, 2013.
- [30] H. Wang, G. Yen, and G. Luo, “Many-objective particle swarm optimization using two-stage strategy and parallel cell coordinate system,” *Cybernetics, IEEE Transactions on*, 2016.
- [31] E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.
- [32] J. Bader and E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [33] R. Hernández Gómez and C. A. Coello Coello, “Improved metaheuristic based on the r2 indicator for many-objective optimization,” in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 679–686.
- [34] M. Wagner and F. Neumann, “A fast approximation-guided evolutionary multi-objective algorithm,” in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2013, pp. 687–694.
- [35] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 468–486.
- [36] B.-Y. Qu, P. Suganthan, and S. Das, “A distance-based locally informed particle swarm model for multimodal optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 3, pp. 387–402, 2013.
- [37] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, “Mommop: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems,” *Cybernetics, IEEE Transactions on*, vol. 45, no. 4, pp. 830–843, 2015.
- [38] J. Horn, N. Nafpliotis, and D. E. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on*. Ieee, 1994, pp. 82–87.
- [39] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, “Reference point based multi-objective optimization using evolutionary algorithms,” *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.
- [40] M. Li, S. Yang, and X. Liu, “Bi-goal evolution for many-objective optimization problems,” *Artificial Intelligence*, vol. 228, pp. 45–65, 2015.
- [41] R. Morgan and M. Gallagher, “Sampling techniques and distance metrics in high dimensional continuous landscape analysis: Limitations and improvements,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 456–461, 2014.
- [42] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 2012, vol. 12.
- [43] Y. Jin, T. Okabe, and B. Sendhoff, “Adapting weighted aggregation for multiobjective evolution strategies,” in *Evolutionary Multi-criterion Optimization*. Springer, 2001, pp. 96–110.
- [44] I. Y. Kim and O. De Weck, “Adaptive weighted-sum method for bi-objective optimization: Pareto front generation,” *Structural and Multidisciplinary Optimization*, vol. 29, no. 2, pp. 149–158, 2005.
- [45] H. Ishibuchi, N. Akedo, and Y. Nojima, “A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems,” in *Learning and Intelligent Optimization*. Springer, 2013, pp. 231–246.
- [46] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Lin, “MOEA/D for flowshop scheduling problems,” in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 2008, pp. 1433–1438.
- [47] A. López-Jaimes and C. A. C. Coello, “Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems,” *Information Sciences*, vol. 277, pp. 1–20, 2014.
- [48] F. Goulart and F. Campelo, “Preference-guided evolutionary algorithms for many-objective optimization,” *Information Sciences*, vol. 329, pp. 236–255, 2016.
- [49] U. K. Wickramasinghe, R. Carrese, and X. Li, “Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm,” in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.
- [50] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer, “A multi-objective approach for workflow scheduling in heterogeneous environments,” in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE Computer Society, 2012, pp. 300–309.
- [51] H. Sato, “Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization,” in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*. ACM, 2014, pp. 645–652.
- [52] K. Sadeghzadeh and M. B. Salehi, “Mathematical analysis of fuel cell strategic technologies development solutions in the automotive industry by the topsis multi-criteria decision making method,” *International Journal of Hydrogen Energy*, vol. 36, no. 20, pp. 13 272–13 280, 2011.
- [53] A. Hatami-Marbini and M. Tavana, “An extension of the electre i method for group decision-making under a fuzzy environment,” *Omega*, vol. 39, no. 4, pp. 373–386, 2011.
- [54] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.
- [55] L. Muflikhah and B. Baharudin, “Document clustering using concept space and cosine similarity measurement,” in *Computer Technology and Development, 2009. ICCTD’09. International Conference on*, vol. 1. IEEE, 2009, pp. 58–62.
- [56] J. Ye, “Cosine similarity measures for intuitionistic fuzzy sets and their applications,” *Mathematical and Computer Modelling*, vol. 53, no. 1, pp. 91–97, 2011.
- [57] H. Wang and X. Yao, “Corner sort for Pareto-based many-objective optimization,” *Cybernetics, IEEE Transactions on*, vol. 44, no. 1, pp. 92–102, 2014.
- [58] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization*. Springer London, 2005, pp. 105–145.
- [59] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, 2006.
- [60] H. Ishibuchi, H. Masuda, and Y. Nojima, “A study on performance evaluation ability of a modified inverted generational distance indicator,” in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 695–702.
- [61] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.
- [62] H. K. Singh, A. Isaacs, and T. Ray, “A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.
- [63] R. Wang, Q. Zhang, and T. Zhang, “Decomposition based algorithms using pareto adaptive scalarizing methods,” *Evolutionary Computation, IEEE Transactions on*, 2016.