

Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization

Miqing Li, Shengxiang Yang, *Senior Member, IEEE*, and Xiaohui Liu

Abstract—It is known that Pareto dominance has its own weaknesses as the selection criterion in evolutionary multiobjective optimization. Algorithms based on Pareto criterion (PC) can suffer from problems such as slow convergence to the optimal front and inferior performance on problems with many objectives. Non-Pareto criterion (NPC), such as decomposition-based criterion and indicator-based criterion, has already shown promising results in this regard, but its high selection pressure may lead to the algorithm to prefer some specific areas of the problem’s Pareto front, especially when the front is highly irregular. In this paper, we propose a bi-criterion evolution (BCE) framework of the PC and NPC, which attempts to make use of their strengths and compensates for each other’s weaknesses. The proposed framework consists of two parts: PC evolution and NPC evolution. The two parts work collaboratively, with an abundant exchange of information to facilitate each other’s evolution. Specifically, the NPC evolution leads the PC evolution forward and the PC evolution compensates the possible diversity loss of the NPC evolution. The proposed framework keeps the freedom on the implementation of the NPC evolution part, thus making it applicable for any non-Pareto-based algorithm. In the PC evolution, two operations, population maintenance and individual exploration, are presented. The former is to maintain a set of representative nondominated individuals and the latter is to explore some promising areas that are undeveloped (or not well-developed) in the NPC evolution. Experimental results have shown the effectiveness of the proposed framework. The BCE works well on seven groups of 42 test problems with various characteristics, including those in which Pareto-based algorithms or non-Pareto-based algorithms struggle.

Index Terms—Bi-criterion evolution (BCE), evolutionary multiobjective optimization (EMO), non-Pareto criterion (NPC), Pareto criterion (PC).

I. INTRODUCTION

THE AREA of multiobjective optimization has developed rapidly over the past few decades, reflecting the need for

Manuscript received December 9, 2014; revised April 8, 2015, July 2, 2015, and September 18, 2015; accepted November 23, 2015. Date of publication December 4, 2015; date of current version September 30, 2016. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) of the U.K. under Grant EP/K001310/1, in part by the National Natural Science Foundation of China under Grant 71110107026 (Major International Joint Research Project) and Grant 61273031, and in part by the EPSRC Industrial Case under Grant 11220252. (*Corresponding author:* Shengxiang Yang.)

M. Li and X. Liu are with the Department of Computer Science, Brunel University London, London UB8 3PH, U.K.

S. Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEVC.2015.2504730

simultaneously dealing with multiple objectives in real-world problems. Unlike global optimization in which there is often a single optimal solution, multiobjective optimization involves a set of Pareto-optimal solutions. Evolutionary algorithms (EAs) have shown high practicability in solving such multiobjective optimization problems (MOPs). Their population-based search aims at finding a finite-size set of well-converged, well-distributed solutions, each representing a unique trade-off among the objectives.

In evolutionary multiobjective optimization (EMO), the selection criterion of individuals in the population plays a key role. Since the output of an EMO algorithm for an MOP is a set of Pareto nondominated solutions, Pareto dominance naturally becomes a viable criterion to select individuals during the evolutionary process. Pareto dominance reflects the weakest assumption about the preference of a decision maker; an individual x is said to Pareto dominate an individual y if it is as good as y in all objectives and better in at least one objective. This criterion, however, fails to distinguish between individuals when they have their own advantage in different objectives of an MOP. In this case, most Pareto-based EMO algorithms, such as the nondominated sorting genetic algorithm II (NSGA-II) [14], introduce the density information of individuals in the population to further rank them, serving the purpose of evolving toward different parts of the problem’s Pareto front.

Despite its popularity in the EMO community, the Pareto criterion (PC) or a Pareto-based algorithm is known to suffer from some drawbacks, such as slow convergence to the optimal front [63], no information on the quantitative difference between two individuals [5], [71], and inferior performance on MOPs with a complex Pareto set (PS) [44] or a high-dimensional objective space [31], [49], [68]. Recently, some non-Pareto selection criteria have been shown to be promising in tackling MOPs. Typically, they convert an objective vector into a scalar value, thus providing a totally ordered set of individuals in the population. Compared with the PC, such criteria have clear advantages, e.g., providing higher selection pressure toward the Pareto front [7], [35], [39] and being easier to work with local search techniques that stem from global optimization [5], [43].

The indicator-based EA (IBEA) [82] and decomposition-based multiobjective EA (MOEA/D) [75] are two representative examples in using the non-Pareto criterion (NPC) to deal with MOPs. IBEA adopts a performance indicator to optimize a desired property of the evolutionary population, and MOEA/D decomposes an MOP into a set of scalar

subproblems and handles them collaboratively with the aid of the information from their neighbors. These two algorithms have laid the foundation for much state-of-the-art work to date, leading to indicator-based and decomposition-based criteria, along with the PC, to have become three mainstream selection criteria in the EMO area [12], [77].

However, an NPC also comes with some shortcomings. Ideally, the outcome of an EMO algorithm is a set of uniformly distributed solutions on the whole Pareto front. These solutions are Pareto-optimal and are supposed to be incomparable in terms of proximity (convergence). However, most non-Pareto criteria, which typically provide higher selection pressure than the PC, make Pareto-optimal solutions comparable, completely [13], [82] or partly [17], [42]. In such criteria, different parts of the Pareto front are treated differently, e.g., the knee and border of the front being usually preferred by the hypervolume (HV)-based criterion [3], [59]. In this way, even some Pareto-optimal solutions may be eliminated during the evolutionary process because they are not in favor with the criterion used, which can result in the final solutions distributed “regularly” but not uniformly along the Pareto front.

Note that the decomposition-based criterion seems to be exempt from the above problem since by using a set of weight vectors, it specifies multiple search directions toward different parts of the Pareto front. One key issue in decomposition-based EMO techniques, however, is how to maintain the uniformity of intersection points of the specified search directions and the problem’s Pareto front. Uniformly distributed weight vectors cannot guarantee the uniformity of the intersection points. In fact, it is very challenging for decomposition-based algorithms to access a set of the well-distributed intersection points for any MOP, in particular in real-world scenarios in which the information of a problem’s Pareto front is often unknown. Although much effort has been made on this issue recently [2], [16], [21]–[24], [37], [58], [72], it is still far from being resolved completely, especially when facing an MOP with a highly irregular optimal front (e.g., a discontinuous or degenerate front).

Given the above, one question could arise: Is it possible to develop an algorithm of synthesizing the Pareto and non-Pareto selection criteria, which makes full use of their advantages as well as effectively avoiding their disadvantages? In this paper, we make an attempt along this line and present a bi-criterion evolution (BCE) framework for MOPs. In BCE, the PC and NPC collaborate, trying to guide the population evolving fast toward the optimal front while simultaneously maintaining the individuals’ diversity during the evolutionary process.

BCE manipulates two evolutionary populations called the NPC population and the PC population, each of which is associated with one criterion. The NPC population steers the PC population that searches toward the optimal front while the PC population compensates the possible diversity loss of the NPC population by exploring some undeveloped (or not well-developed) but potentially promising regions in the objective space. The two populations communicate with each other in a generational manner; once one population produces good

individuals, the other is able to apply them directly within its search process.

BCE keeps it free on the design of the NPC evolution part, thus making the framework applicable for any non-Pareto-based EMO algorithm in the area. Effort of BCE is primarily on the PC evolution part. In the PC evolution, an individual exploration operation, coupled with a novel population maintenance strategy, is proposed to adaptively allocate resources (search effort) based on the information contrast between the current states of the two evolution parts.

The rest of this paper is organized as follows. Section II explains the motivation of the proposed approach. Section III is devoted to the description of BCE, including the basic algorithmic framework, the population maintenance and individual exploration operations, and the analysis of the algorithm’s time complexity. Section IV experimentally verifies the proposed BCE framework, based on its implementation with three representative non-Pareto-based algorithms. Further investigation and discussion of BCEs behavior are given in Sections V and VI, respectively. Finally, Section VII draws the conclusion of this paper.

II. MOTIVATION AND RELATED WORK

Over the past few years, non-Pareto criteria have demonstrated their success in dealing with many challenging MOPs, such as an MOP with a huge number of local Pareto fronts [17], with a complex PS [44], [74], or with a high-dimensional objective space [32], [51], [68]. They typically provide higher selection pressure than the PC by either modifying the traditional Pareto dominance relation (such as the ϵ -dominance [17], [42], [67], fuzzy-based dominance [26], and dominance area control [60]) or introducing a quantitative individual comparison criterion (such as the distance-based criterion [54], [70], indicator-based criterion [8], [39], [82], and decomposition-based criterion [55], [75]).

However, non-Pareto criteria also suffer from problems, e.g., in terms of maintaining individuals’ diversity (especially uniformity) in the population. In general, the ideal output of an EMO algorithm, in the absence of any preference information, is a set of uniformly distributed nondominated solutions over the whole Pareto front. This means that the comparison between the Pareto-optimal solutions should be based solely on their density information. However, this is not the case in non-Pareto criteria whereby the Pareto optimal solutions could be ranked, depending not only on their density but also on their position in the population as well as the shape of the Pareto front. For example, the ϵ -dominance criterion [42] is likely to eliminate boundary individuals of the population [27], [51]. Some indicator-based criteria, such as the HV [79] and R2 [9], prefer the knee region of the Pareto front [19], [59]. The algorithms based on the decomposition criterion search toward a set of points intersected by the specified search directions and the Pareto front but struggle to maintain the uniformity of these intersection points when the front is highly irregular [24], [37], [58].

Next, we give an empirical example to show the failure of a non-Pareto-based algorithm in providing a set of

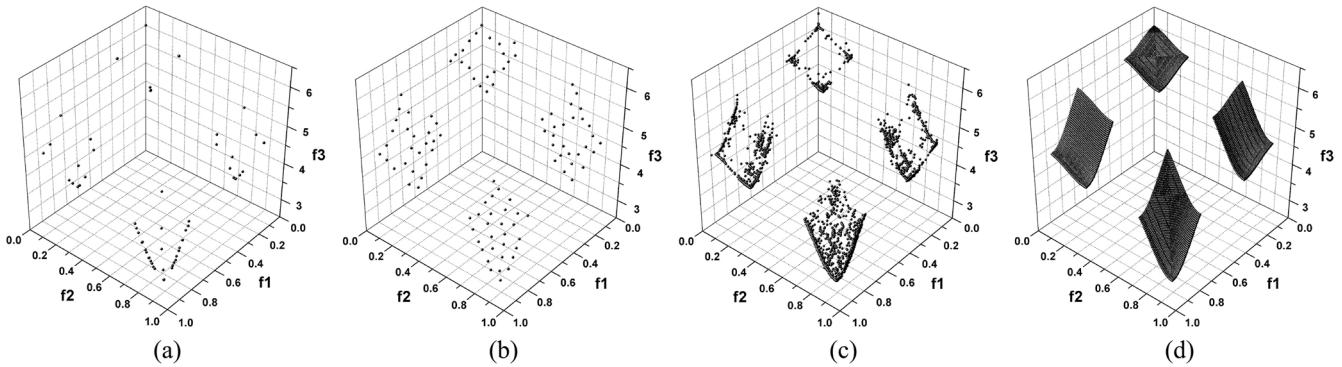


Fig. 1. Empirical example of the failure of an NPC in both diversity maintenance and search in which the results are obtained with respect to one run of MOEA/D+TCH on the problem DTLZ7. (a) Solution set maintained by the original criterion of MOEA/D+TCH. (b) Solution set maintained by the criteria of Pareto dominance and density. (c) Nondominated set of all solutions produced in the run. (d) Pareto front.

representative solutions. Fig. 1 shows the results with respect to one typical run¹ of a popular decomposition-based algorithm, MOEA/D [75] with the Tchebycheff (TCH) scalarizing function² (denoted as MOEA/D+TCH), on a discontinuous test problem DTLZ7 [18]. The final solutions obtained by MOEA/D+TCH are plotted in Fig. 1(a). For contrast, Fig. 1(b) gives the final result of the solution set maintained by the criteria of Pareto-based algorithms (i.e., solutions being tested first by their Pareto dominance relation and then by their density³) in this run of MOEA/D+TCH. That is, an external archive set is added in the algorithm to store well-distributed nondominated solutions produced throughout the whole evolutionary process. In addition, the nondominated set of all solutions produced in this run is given in Fig. 1(c).

As can be seen from Fig. 1(a) and (c), MOEA/D+TCH fails to select a set of diverse solutions from all the solutions produced in the whole evolutionary process. In contrast, the selection criteria of Pareto-based algorithms, which consider the Pareto dominance relation and density of candidate solutions, can make the algorithm's output representative, as shown in Fig. 1(b). On the other hand, the deficiency of the algorithm in diversity maintenance also has a detrimental effect on its search ability. To explain this, the Pareto front of the problem is added in Fig. 1(d) for the comparison between the real optimal solutions and the solutions produced during the evolutionary process. From Fig. 1(c) and (d), it can be observed that there exist several large pieces of unexplored regions in the MOEA/D+TCHs search process. This occurrence can be attributed to the fact that the selection operation in this non-Pareto algorithm is always around some particular points [Fig. 1(a)] at each generation, thus leading individuals' exploration to concentrate only on some specific regions of the objective space.

The above problems of non-Pareto criteria are precisely the underlying motivation of this paper. In this paper, we introduce

¹The parameter setting in the run is the same as in the experimental studies, described in Section IV.

²In order to obtain more uniform solutions, in the TCH scalarizing function, “multiplying the weight vector w_i ” in the original MOEA/D+TCH [75] is replaced by “dividing w_i ,” as suggested and practiced in recent studies [16], [45].

³Here, individuals' density is estimated by the method in BCE that is described in Section III-B.

a BCE framework of Pareto and non-Pareto criteria in order to use their strengths and compensate for each other's weaknesses.

It is worth pointing out that the combination of NPC and PC is not uncommon in EMO. For example, Ishibuchi *et al.* [33] combined the Pareto-based algorithm NSGA-II with the weighted-sum criterion to probabilistically pick out solutions in both mating and environmental selection processes. Al Moubayed *et al.* [1] used a decomposition-based criterion to select the leaders in multiobjective particle swarm optimization and introduced the crowding distance to maintain the diversity of nondominated solutions in the decision and objective spaces. Deb and Jain [16] proposed a hybrid EMO algorithm, NSGA-III, which uses the Pareto nondominated sorting to develop convergence and the decomposition-based criterion to maintain diversity during the evolutionary process.

On the other hand, some studies in the literature adopted multiple archives (or populations) to separately promote convergence and diversity during the evolutionary process. Wang *et al.* [69] developed a two-archive many-objective algorithm, with one archive being driven by an indicator-based criterion and the other being maintained by an L_p -norm-based distance criterion. Zăvoianu *et al.* [73] presented a hybrid co-EA with three populations, each one associated with a classic algorithm, i.e., SPEA2 [78], differential evolution (DE) [41], and a decomposition-based algorithm. Cai *et al.* [10] proposed a hybrid EMO algorithm for combinatorial MOPs, by using a decomposition-based strategy to guide its internal population and a domination-based sorting technique to maintain the external archive. In addition, the idea of having separate archives has also been used in multiobjective scatter search whereby the reference set is split into two subsets that promote convergence and diversity. In multiobjective scatter search algorithms, Pareto dominance and decomposition criteria are often used in the convergence-promoting subset and distance-based criteria in the diversity-promoting subset [6], [53], [56].

An important difference between the proposed BCE and existing hybrid EMO algorithms with multiple criteria and/or multiple archives is that BCE takes advantage of the information contrast between the evolutionary populations based on distinct selection criteria, thus making the search focused

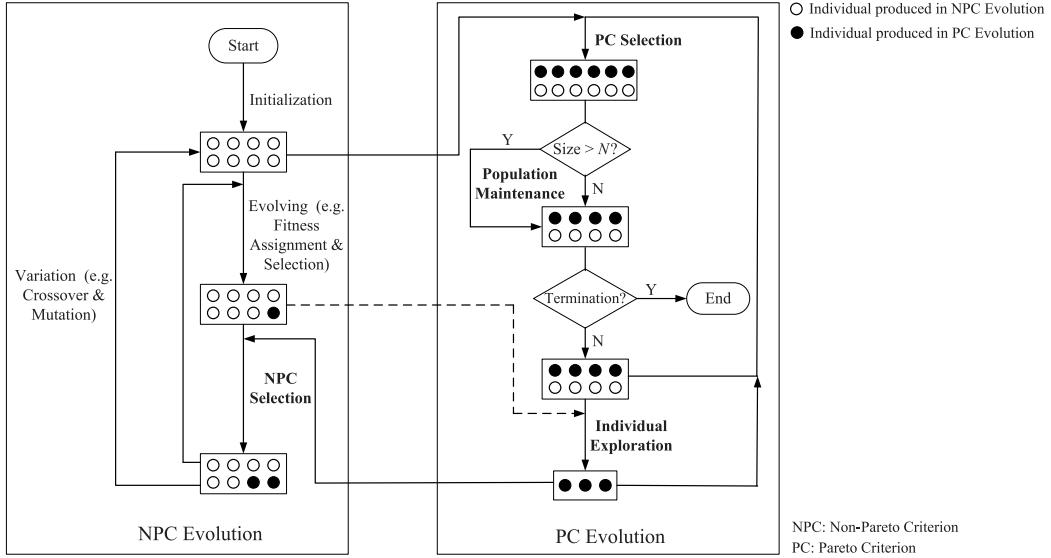


Fig. 2. Overall framework of BCE.

on promising regions in terms of both Pareto and non-Pareto criteria. Another clear difference is that BCE is a general framework rather than a specific algorithm, and it can work with any non-Pareto EMO algorithm.

III. BI-CRITERION EVOLUTION

Fig. 2 gives the overall framework of BCE. As shown, BCE consists of two evolution parts: NPC evolution and PC evolution. BCE keeps the freedom on the implementation of the NPC evolution part—any non-Pareto EMO algorithm can be directly embedded, with all components (population setting, individual initialization, fitness assignment, selection, variation, etc.) remaining unchanged. The only newly introduced operation is that the population for the next-generation evolution (the bottom box) comprises individuals that are selected from itself and newly produced individuals in the PC evolution part (called the NPC selection).

For the PC evolution part, the manipulated population (i.e., the PC population) only preserves the Pareto nondominated solutions produced in both NPC and PC evolution, thereby having a varying size. When the size of the PC population is larger than a predefined threshold, a population maintenance operation will be implemented to eliminate some poorly distributed individuals. If the termination condition here is satisfied (e.g., reaching a preset number of evaluations), the evolution ends with the PC population as the final output. Otherwise, an individual exploration operation is implemented to explore some promising individuals in the PC population, bearing the evolutionary information from the NPC population.

In BCE, the two populations share and exchange information frequently but evolve based on their own criterion. Any new individual (wherever it is produced) will be considered in both sides of BCE to see if it could be preserved in their own population. In general, the PC population can be regarded as a good complement to the NPC population. It is able to not

only preserve representative Pareto nondominated solutions that could be eliminated in the NPC evolution but also reflect the current status of the NPC evolution by the information contrast between the two populations.

Next, we describe key operations of BCE. They are the PC and NPC selection, population maintenance, and individual exploration.

A. PC Selection and NPC Selection

As their names suggest, the PC and NPC selections are to select individuals (from the considered population and the newly produced individuals) according to the PC and NPC, respectively. The PC selection is implemented by directly picking out the Pareto nondominated individuals from the mixed set of the PC population and new individuals produced in both the NPC and the PC evolutions.

The NPC selection, in general, can be simply implemented by the environmental selection operation of the embedded non-Pareto-based algorithm. For example, in the NPC selection of BCE-IBEA (i.e., BCE with IBEA embedded into its NPC evolution part), the resulting NPC population comprises the individuals with the highest fitness with respect to the considered criterion (indicator) in the mixed set of the NPC population and the new individuals from the PC evolution. However, this is impracticable for some algorithms for which the survival of a newly produced individual is relevant to the information from its parent(s), such as MOEA/D. This is because the candidate individuals in the NPC selection are from different evolution parts, without being in the parent-child relationship. For these algorithms, the NPC selection compares each individual from the PC evolution with all the members of the NPC population. If an individual from the PC evolution performs better than one or more population members with respect to the considered criterion, then it replaces one of them (chosen at random); otherwise, it is discarded. Algorithm 1 gives the procedure of the NPC selection.

Algorithm 1 NPC Selection (Q, T)

Require: Q (non-Pareto criterion population), T (newly produced individual set in the Pareto criterion evolution part), $label$ (type of the environmental selection in the embedded non-Pareto algorithm)

- 1: $label \leftarrow SelectionType()$ /* Return 1 if the survival of a newly produced individual is irrelevant to its parent(s) in the selection process of the non-Pareto algorithm; otherwise, return 0 */
- 2: **if** $label = 1$ **then**
- 3: $Q \leftarrow EnvironmentalSelection(Q, T)$
 /* Select $|Q|$ individuals with the highest fitness with respect to the considered non-Pareto criterion from $Q \cup T$ */
- 4: **else**
- 5: **for all** $t \in T$ **do**
- 6: $Z \leftarrow \emptyset$
- 7: **for all** $q \in Q$ **do**
- 8: **if** t is better than q with respect to the considered criterion **then**
- 9: $Z \leftarrow Z \cup q$
- 10: **end if**
- 11: **end for**
- 12: **if** $Z \neq \emptyset$ **then**
- 13: $z \leftarrow Random(Z)$
 /* Select one individual from Z at random */
- 14: $Q \leftarrow Q \setminus z$
- 15: $Q \leftarrow Q \cup t$
- 16: **end if**
- 17: **end for**
- 18: **end if**
- 19: **return** Q

B. Population Maintenance

In the PC evolution part, the population preserves the Pareto nondominated individuals produced during the whole search process and varies in size. When the size of the population exceeds a predefined capacity, population maintenance will be activated to truncate some of its individuals with poor distribution. It is known that an effective population maintenance operation can maintain a set of representative individuals, which is independent of the properties of the problem (e.g., the number of objectives and the shape of the Pareto front). In this paper, we present a niche-based approach, attempting to preserve a set of representative individuals for any MOP.

Niching is a class of popular diversity maintenance techniques in the EA field. Originating from the idea of sharing resources, niching can be used to measure individuals' crowding degree (density) in the population. Here, we estimate the crowding degree of an individual by considering both the number and the location of the individuals in its niche. Specifically, the crowding degree of an individual p in the population P is defined as

$$D(p) = 1 - \prod_{q \in P, q \neq p} R(p, q) \quad (1)$$

$$R(p, q) = \begin{cases} d(p, q)/r, & \text{if } d(p, q) \leq r \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

where $d(p, q)$ denotes the Euclidean distance between individuals p and q , and r is the radius of the niche (its setting will be explained later). Note that the scale of the problem's objectives could be highly different and this will affect the estimation of individuals' crowding degree. To avoid this kind of problem,

in BCE, all the objectives will be normalized (with respect to their minimum and maximum values in the population) when the considered operation involves the integration of multiple objectives.

Next, we give some explanations of the proposed crowding degree estimation method.

- 1) The crowding degree of an individual is in the range $[0, 1]$, with a lower value being preferable. An individual having the crowding degree 0 means that there is no other individual in its niche. On the other hand, duplicate individuals have the highest crowding degree 1, regardless of the distribution of other individuals in their niche.
- 2) The crowding degree of an individual is determined by the number of its neighbors (i.e., the individuals in its niche) and the distance between it and these neighbors. Individuals having more neighbors or closer distance to their neighbors are likely to obtain a higher (worse) crowding degree.
- 3) The crowding degree of an individual is influenced more by its closer neighbor(s). For example, considering two individuals p and q , let both have two neighbors and the sum of the distance to their own neighbors be the same (say 0.2 and 0.8 for p and 0.4 and 0.6 for q). According to the definition, p , which has a shorter distance (0.2) to its closer neighbor, will have a higher crowding degree than q ($1 - 0.16/r^2 = 0.84 > 1 - 0.24/r^2 = 0.76$, assuming $r = 1.0$). Actually, even if p has only one neighbor (closer one), its crowding degree is still higher than that of q ($1 - 0.2/r = 0.8 > 1 - 0.24/r^2 = 0.76$). This means that an individual that has a very close neighbor will be assigned a high crowding degree no matter how far it is from other individuals in the population. This is in line with the target of developing the diversity of individuals.

One crucial issue in the proposed crowding degree estimator is the setting of the niche radius, which determines the number of neighbors as well as their location in the niche. Unlike some niching techniques in which it is fixed and/or set by the user, the niche radius in the proposed estimator is determined by the evolutionary population. We consider the average of the distance from all the individuals to their k th nearest individual in the population as the radius, attempting to enable most of the individuals to have one or several neighbors in their niche. Here, k is set to 3. The reason of this setting will be explained in detail in the discussion section of this paper (Section VI).

Based on the crowding degree of individuals in the population, the truncation operation can be simply implemented. First, the individual that has the highest crowding degree is removed; if there are several individuals with the highest crowding degree, the tie will be split randomly. Then, the crowding degree of the individuals who are neighbors of the removed individual (i.e., in its niche) is renewed, and, again, the current most crowded individual is found and removed. This process is repeated until a predefined population size is achieved. Overall, the proposed method iteratively removes crowded individuals and thus leaves a

representative population, which can also be observed in the example of Fig. 1 [see Figs. 1(b) and (c)].

C. Individual Exploration

In BCE, the NPC evolution generally has higher selection pressure than the PC evolution and may sometimes prefer partial area(s) of the Pareto front. This may cause repeating searches on some particular regions of the objective space. The individual exploration operation in this section aims to cover this issue. It attempts to explore some promising individuals in the PC population that have been eliminated, are not well-developed, or are even unvisited in the NPC evolution. This exploration is adaptive, based on the information comparison between the two evolutionary populations. If the NPC population has been found to be well distributed, little exploration will be made; otherwise, much exploration will be made around those promising individuals.

Now, a key question may arise—what individuals are promising and need to be explored? Since the PC population is comprised of a set of representative nondominated individuals, it generally performs well in both convergence and diversity. Nevertheless, it is unnecessary to explore the whole PC population because some of its individuals may already be well explored in the NPC evolution. Such individuals are preferred by the considered NPC, and there may be many individuals in the NPC population located around the regions where such individuals reside.

In view of this, we consider two kinds of individuals out of the whole PC population: 1) individuals whose niche has no NPC individual⁴ and 2) individuals whose niche has only one NPC individual. The first kind of individuals is clearly not preferred by the considered NPC. Exploring them means to probe into undeveloped regions in the NPC evolution. The niches in which the second kind of individuals resides correspond to low-density regions of the NPC population. Exploring them means to probe into the regions that are not well developed in the NPC evolution but may be potentially promising since they still have individual(s) existing in both the NPC and the PC populations after (iterative) selection based on the non-Pareto and Pareto criteria, respectively.

Algorithm 2 gives the main procedure of individual exploration. As shown, the algorithm can primarily be divided into two parts. One is to determine which individuals in the PC population will be explored (steps 3–13) and the other is to carry out the exploration on those individuals (steps 15–18). In the proposed framework, the variation operation (step 16) is not fixed and can be freely specified by users. It can be the same with what is in the NPC evolution (as done in our experimental studies), be chosen from other existing variation operators, or even be directly designed for the exploration. In addition, note that in different variation operators the number of parent individuals may be different. For a variation operator with only one parent (like mutation), the explored individual is applied directly. For a variation operator with two or more parents (like crossover), the explored individual is considered

⁴For brevity, individuals in the NPC and PC populations are denoted as NPC and PC individuals, respectively.

Algorithm 2 Exploration (P, Q)

```

Require:  $P$  (Pareto criterion population),  $Q$  (non-Pareto criterion population),  $S$  (set of the individuals to be explored),  $T$  (set of newly produced individuals)
1:  $S \leftarrow \emptyset$ 
2:  $r \leftarrow \text{Radius}()$  /* Determine the size of the niche */
3: for all  $p \in P$  do
4:    $count \leftarrow 0$  /* For recording the number of the NPC individuals in the niche of  $p$  */
5:   for all  $q \in Q$  do
6:     if  $d(p, q) \leq r$  then
7:        $count \leftarrow count + 1$  /* When  $q$  is in the niche of  $p$  */
8:     end if
9:   end for
10:  if  $count = 0$  or  $count = 1$  then
11:     $S \leftarrow S \cup p$ 
12:  end if
13: end for
14:  $T \leftarrow \emptyset$ 
15: for all  $s \in S$  do
16:    $s' \leftarrow \text{Variation}(s)$ 
17:    $T \leftarrow T \cup s'$ 
18: end for
19: return  $T$ 

```

as one parent (or the primary parent in the operator, e.g., in DE) and the remaining parent(s) will be selected randomly from the PC population.

In step 2 of Algorithm 2, the radius of the considered niche is calculated. The niche range is an important factor in individual exploration, which, together with the distribution of NPC individuals, determines how many individuals will be explored in the PC population. A small enough niche is likely to lead all PC individuals to be explored, and a large enough niche can cause none of them to be explored. Here, we introduce a variable niche whose range varies with the size of the PC population.

The PC population preserves only nondominated individuals, and its size can reflect the role of the Pareto dominance criterion during the evolutionary process. A small population size means that Pareto dominance can provide sufficient selection pressure to eliminate poorly performed individuals. This usually happens in the initial stage of the evolution. At this time, the population maintenance operation is not activated, and the PC population that stores all nondominated individuals produced in both the NPC and the PC evolutions represents the best individuals found so far. Therefore, it is desirable to put more effort into exploring it. With the progress of the evolution, more and more individuals are produced, and Pareto dominance may gradually fail to provide sufficient selection pressure. When newly produced nondominated individuals significantly exceed the remaining slots of the population capacity, the PC evolution will slow down. At this time, it is beneficial to make relatively less exploration on the PC population, thus leading to more resources possessed by the NPC evolution, which generally has high selection pressure. Given the above, the radius of the niche is determined as

$$r = (N'/N) \times r_0 \quad (3)$$

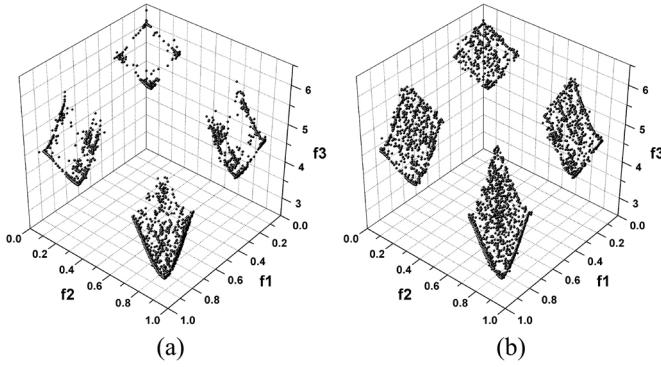


Fig. 3. Nondominated set of all the solutions produced in one run of (a) MOEA/D+TCH and (b) BCE-MOEA/D+TCH on DTLZ7, respectively.

where N denotes the capacity of the PC population, N' denotes the actual size of the PC population before the truncation, and r_0 is the basic niche radius, calculated in the same way as in the population maintenance operation.

In BCE, individual exploration in the PC evolution inevitably competes with the variation operation in the NPC evolution for limited computational resources (i.e., function evaluations). Given a fixed computational budget, the number of individuals explored directly affects the evolutionary level of the NPC population. However, it is worth noting that individual exploration here is adaptive, depending on the current evolutionary status of the NPC population. When the NPC population has diversity loss [like the case in Fig. 1(a) in which the decomposition-based criterion struggles to maintain diversity], intensive exploration will be made. When the population has been found to be well distributed, little or even no exploration will be done; for instance, for the test function DTLZ2 [18] the decomposition-based evolutionary population can work very well and thus no individual is explored in the PC population (this will also be empirically presented in Section V-B).

Finally, Fig. 3 gives the comparative results of the original MOEA/D+TCH [i.e., Fig. 1(c)] and BCE-MOEA/D+TCH (BCE with MOEA/D+TCH embedded into its NPC evolution part) by plotting all of their nondominated individuals produced in one run. In contrast to MOEA/D+TCHs solutions that are located around some specific regions, the solutions produced by BCE-MOEA/D+TCH nearly cover the whole optimal space. This difference can be fully attributed to the individual exploration operation in the PC evolution part of the algorithm, which conducts the search on some undeveloped (or not well-developed) regions in the NPC evolution.

D. Computational Complexity of One Generation of BCE

The computational cost of BCE comes from the NPC evolution and the PC evolution. For simplicity, let both parts have the population size (capacity) N . For the time complexity of the NPC evolution, there are two possible situations, depending on the selection operation in the embedded non-Pareto algorithm. When the survival of an individual is determined by its fitness in the population (like in IBEA),

the NPC selection is implemented in the same way as the individual selection in the embedded algorithm (see Section III-A). In this case, the NPC evolution has the same time complexity as the embedded algorithm (denoted as C). On the other hand, when the survival of an individual is relevant to the information from its parents (like in MOEA/D), the NPC selection is implemented by comparing the individuals produced in the PC evolution with the members of the NPC population. This requires $O(N^2)$ comparisons at most. Hence, the time complexity of the NPC evolution in this situation is C or $O(N^2)$, whichever is larger.

The computational cost of the PC evolution part is determined by three operations: the PC selection, population maintenance, and individual exploration. The PC selection, which identifies nondominated individuals from a population with $3N$ members at most, requires $O(mN^2)$ comparisons [14], where m is the number of objectives. In the population maintenance, the Euclidean distance between each pair of individuals in the population is first calculated, which requires $O(mN^2)$ computations. Then, determining the niche radius requires $O(N^2)$ computations in which finding the k th smallest distance ($k = 3$) for an individual needs $O(N)$ comparisons. Thereafter, the crowding degree estimation and the population truncation are sequentially implemented. Both require $O(N^2)$ computations (or comparisons). It is worth mentioning that in the population truncation, we need only to update the crowding degree of the neighbors of the removed individual (i.e., the individuals that are in the same niche of the removed individual). In general, the niche of an individual only has a few individuals (independent of N) due to the setting of the radius (namely, the average distance from all individuals in the population to their third nearest individual). In the individual exploration, the Euclidean distance between the individuals and the radius of the niche are also calculated first, which require $O(mN^2)$ and $O(N^2)$ computations, respectively. Then, determining which individuals in the population will be explored requires $O(N^2)$ comparisons (steps 3–13 in Algorithm 2). Finally, carrying out the exploration operation on the selected individuals requires $O(N)$ computations at most. Therefore, the total time complexity of the PC evolution is $O(mN^2)$.

To summarize, the overall computational complexity of one generation of BCE is bounded by C or $O(mN^2)$, whichever is larger, where C is the computational complexity of the embedded non-Pareto algorithm.

IV. PERFORMANCE VERIFICATION OF BCE

The proposed framework is verified by embedding non-Pareto EMO algorithms into its NPC evolution part and comparing these non-Pareto algorithms with the resulting BCE algorithms. We consider the two representative non-Pareto-based algorithms IBEA [82] and MOEA/D [44], which lead the evolution via the indicator-based criterion and decomposition-based criterion, respectively. In MOEA/D, two scalarizing functions TCH and penalty-based boundary intersection (PBI) are commonly used in the literature, and both are included in our experiments in view of their good performance for different

TABLE I
SETTINGS AND PROPERTIES OF TEST PROBLEMS. m AND d DENOTE THE NUMBER OF OBJECTIVES AND DECISION VARIABLES, RESPECTIVELY

Problem	m	d	Properties	Problem	m	d	Properties	Problem	m	d	Properties
SCH1	2	1	Convex	WFG7	2	22	Concave, Biased	UF2	2	30	Convex, Complex PS
SCH2	2	1	Discontinuous	WFG8	2	22	Concave, Nonseparable, Biased	UF3	2	30	Convex, Complex PS
KUR	2	3	Discontinuous	WFG9	2	22	Concave, Nonseparable, Deceptive, Biased	UF4	2	30	Concave, Complex PS
ZDT1	2	30	Convex	VNT1	3	2	Convex	UF5	2	30	Linear, Discrete, Complex PS
ZDT2	2	30	Concave	VNT2	3	2	Mixed	UF6	2	30	Linear, Discontinuous, Complex PS
ZDT3	2	30	Discontinuous	VNT3	3	2	Mixed, Degenerate	UF7	2	30	Linear, Complex PS
ZDT4	2	10	Convex, Multimodal	DTLZ1	3	7	Linear, Multimodal	UF8	3	30	Concave, Complex PS
ZDT6	2	10	Concave, Multimodal, Biased	DTLZ2	3	12	Concave	UF9	3	30	Linear, Discontinuous, Complex PS
WFG1	2	22	Mixed, Biased	DTLZ3	3	12	Concave, Multimodal	UF10	3	30	Concave, Complex PS
WFG2	2	22	Convex, Discontinuous, Nonseparable	DTLZ4	3	12	Concave, Biased	DTLZ2(4)	4	13	Concave
WFG3	2	22	Linear, Degenerate, Nonseparable	DTLZ5	3	12	Concave, Degenerate	DTLZ2(6)	6	15	Concave
WFG4	2	22	Concave, Multimodal	DTLZ6	3	12	Concave, Degenerate, Biased	DTLZ2(10)	10	19	Concave
WFG5	2	22	Concave, Deceptive	DTLZ7	3	22	Mixed, Discontinuous, Multimodal	DTLZ5(2,10)	10	19	Concave, Degenerate
WFG6	2	22	Concave, Nonseparable	UF1	2	30	Convex, Complex PS	DTLZ5(3,10)	10	19	Concave, Degenerate

MOPs [16], [32], [44], [48]. Note that MOEA/D used here is sourced from [44] rather than from its original paper [75]. This improved version can largely enhance the diversity of the population by allowing parent individuals to be selected from the whole population as well as setting a limit of the maximal number of individuals replaced by a newly produced child individual. In addition, in the TCH scalarizing function, we replace “multiplying the weight vector” with “dividing it” for obtaining more uniform individuals, as pointed out in [16] and [37]. Overall, the intention that we consider this version of MOEA/D is to verify the effectiveness of BCE even when the considered non-Pareto algorithms already work fairly well in terms of diversity maintenance on most MOPs.

A comprehensive set of 42 MOPs are introduced in the experiments. These test problems, which are widely used in the area, have various properties, such as having a convex, concave, mixed, discontinuous, or degenerate Pareto front, having a multimodal, biased, or deceptive search space, and/or having strong-linkage decision variables. They certainly include some MOPs for which non-Pareto algorithms generally work well, like an MOP with a linear (or fairly regular) Pareto front, and also have some for which the algorithms may encounter difficulties, like an MOP with a discontinuous (or highly irregular) Pareto front. Table I summarizes the properties and configuration of these MOPs. All the problems are configured as described in their original papers [18], [28], [61], [66], [74], [80].

To compare the performance of the algorithms, two widely used quality indicators, such as the inverted generational distance (IGD) [16], [75] and HV [79], are considered because they can provide a combined information of convergence and diversity of a solution set. IGD measures the average Euclidean distance from uniformly distributed points along the whole Pareto front to their closest solution in the obtained solution set, and a smaller value is preferable. HV calculates the volume of the objective space between the obtained solution set and a specified reference point, and a larger value is preferable.

In the calculation of HV, two crucial issues are the scaling of the search space [20] and the choice of the reference point [3]. Since the objectives in the considered test problems take different ranges of values, we standardize the objective value of the obtained solutions according to the range of the problem’s Pareto front. Following the recommendation in [34],

TABLE II
POPULATION SIZE AND FUNCTION EVALUATIONS IN THE EXPERIMENTS

Test Problems	Population Size	Function Evaluations
2-Obj. UF	600	300 000
3-Obj. UF	1 000	300 000
General 2-Obj. MOPs	100	25 000
General 3-Obj. MOPs	105	30 000
4-Obj. MOPs	220	100 000
6-Obj. MOPs	252	100 000
10-Obj. MOPs	220	100 000

the reference point is set to 1.1 times the upper bound of the Pareto front (i.e., $r = 1.1^m$) to emphasize the balance between proximity and diversity of the obtained solution set. Note that solutions that do not dominate the reference point are discarded (i.e., solutions that are worse than the reference point in at least one objective contribute zero to HV).

All the results presented in this paper are obtained by executing 30 independent runs for each algorithm. For a fair comparison, all the algorithms have the same size (or capacity) of the population (for BCE, this refers to both the NPC and the PC populations) and the same number of function evaluations on each problem. Table II lists the settings of the population size and function evaluations for all the test problems in the experiments. For the UF functions from the CEC2009 competition [74], the population size and function evaluations are specified the same as in their original report [76]. For other MOPs, we used a smaller population size and fewer function evaluations as they are generally easier than the UF functions. Like some existing studies [52], the number of function evaluations is set to 25 000 and 30 000 for two- and three-objective MOPs, respectively. Note that in MOEA/D, the population size corresponds to the number of weight vectors and the algorithm cannot generate uniformly distributed weight vectors at an arbitrary number. So, we set the population size consistent with the number of the uniformly generated weight vectors in MOEA/D. That is 100, 105, 220, 252, and 220 for the 2-, 3-, 4-, 6-, and 10-objective MOPs, respectively. In addition, given that many-objective problems often bring bigger challenges for EMO algorithms than MOPs with two or three objectives [57], we assign them a larger population size and more function evaluations, following the practice in [47].

Parameters need to be set in the considered algorithms. According to [74], the size of the neighborhood, the

TABLE III
IGD RESULTS (MEAN AND SD) OF THE THREE GROUPS OF PAIRED ALGORITHMS. THE BETTER MEAN FOR EACH CASE IS HIGHLIGHTED IN BOLDFACE

Property	Problem	IBEA	BCE-IBEA	MOEA/D+TCH	BCE-MOEA/D+TCH	MOEA/D+PBI	BCE-MOEA/D+PBI
Convex	SCH1	1.9093E-2(4.1E-4)	1.6732E-2(1.1E-4)†	4.9139E-2(1.5E-3)	1.6761E-2(1.4E-4)†	1.7765E-1(5.2E-3)	1.6729E-2(9.3E-5)†
	ZDT1	4.0773E-3(6.8E-5)	3.9475E-3(5.0E-5)†	4.3626E-3(3.4E-4)	4.2756E-3(9.4E-5)†	8.2654E-3(1.7E-3)	5.9401E-3(4.3E-4)†
	ZDT4	6.2464E-1(1.1E-1)	1.2270E-1(5.9E-2)†	1.0001E-2(3.4E-3)	8.2586E-3(3.2E-3)†	3.7084E-2(2.8E-2)	1.3458E-2(6.5E-3)†
	VNT1	1.6688E-1(2.7E-2)	1.2731E-1(3.3E-3)†	2.2398E-1(1.3E-3)	1.2240E-1(2.7E-3)†	1.7364E-1(5.6E-4)	1.2544E-1(2.7E-3)†
Concave	ZDT2	9.1909E-3(3.7E-4)	4.0420E-3(3.3E-4)†	4.3342E-3(1.8E-4)	4.1688E-3(9.3E-5)†	1.0540E-2(1.2E-3)	7.2119E-3(7.8E-4)†
	ZDT6	5.4849E-3(1.8E-4)	3.4449E-3(8.6E-5)†	1.5027E-2(1.7E-3)	1.0328E-2(1.3E-3)†	3.9299E-2(4.6E-3)	2.1787E-2(2.7E-3)†
	WFG4	1.8370E-2(8.3E-4)	1.3005E-2(4.6E-4)†	2.2551E-2(1.8E-3)	1.9664E-2(1.5E-3)†	5.0721E-2(5.2E-3)	3.8183E-2(3.9E-3)†
	WFG5	7.2029E-2(7.6E-4)	6.7165E-2(1.7E-4)†	6.8459E-2(5.9E-4)	6.8558E-2(6.3E-4)	8.2498E-2(2.9E-3)	7.7588E-2(1.8E-3)†
	WFG6	6.5849E-2(9.4E-3)	6.1760E-2(9.2E-3)†	6.8500E-2(1.0E-2)	6.4056E-2(8.2E-3)	1.0092E-1(1.4E-2)	8.6534E-2(1.2E-2)†
	WFG7	2.1525E-2(1.0E-3)	1.4264E-2(3.1E-4)†	1.6770E-2(4.7E-4)	1.5242E-2(3.3E-4)†	3.7482E-2(3.3E-3)	2.9628E-2(2.2E-3)†
	WFG8	9.3529E-2(8.0E-3)	8.0848E-2(6.1E-3)†	6.7532E-2(5.6E-3)	8.5839E-2(5.6E-3)	1.1848E-1(7.9E-3)	1.0653E-1(8.1E-3)†
	WFG9	7.8794E-2(5.2E-2)	7.4934E-2(5.2E-2)†	3.5633E-2(2.4E-2)	5.6954E-2(4.5E-2)†	5.6432E-2(1.6E-2)	7.3108E-2(3.9E-2)†
	DTLZ2	1.1894E-1(2.2E-3)	5.3051E-2(1.0E-3)†	5.1151E-2(4.2E-4)	5.1708E-2(7.7E-4)†	5.0151E-2(3.1E-6)	5.1369E-2(8.3E-4)†
	DTLZ3	5.1829E-1(1.5E-1)	5.3253E-1(4.3E-1)	5.0856E-1(6.5E-1)	1.6158E+0(1.3E+0)†	1.0267E+0(9.1E-1)	1.4062E+0(1.1E+0)
	DTLZ4	4.6424E-1(3.4E-1)	2.3409E-1(3.4E-1)†	1.7346E-1(2.3E-1)	5.2207E-2(9.4E-4)†	1.1261E-1(1.7E-1)	5.1944E-2(9.8E-4)†
Linear	DTLZ5	2.5627E-2(1.1E-3)	4.1949E-3(2.4E-4)†	1.8107E-2(1.5E-5)	4.4611E-3(3.2E-4)†	3.1920E-2(1.0E-4)	4.5173E-3(3.4E-4)†
	DTLZ6	1.0148E-1(2.0E-2)	7.3724E-2(2.5E-2)†	3.9000E-1(3.3E-2)†	6.4541E-1(5.7E-2)	7.6720E-1(4.8E-2)†	
	WFG3	1.5944E-2(1.2E-3)	1.5917E-2(1.8E-3)	2.5079E-2(4.3E-3)	2.3228E-2(3.3E-3)†	4.7442E-2(8.1E-3)	3.9358E-2(7.4E-3)†
Mixed	DTLZ1	1.8267E-1(1.7E-2)	2.2154E-2(2.0E-3)†	1.9571E-2(1.1E-3)	2.1149E-2(1.8E-3)†	1.9130E-2(5.7E-4)	2.2134E-2(2.4E-2)†
	WFG1	7.5883E-1(6.2E-2)	7.7944E-1(6.3E-2)	1.0487E+0(7.2E-2)	9.6279E-1(5.4E-2)†	1.1983E+0(5.9E-2)	1.0490E+0(5.3E-2)†
	VNT2	4.2602E-2(9.4E-3)	1.2219E-2(3.2E-4)†	4.6425E-2(2.9E-4)	1.2121E-2(2.8E-4)†	5.8432E-2(2.4E-4)	1.2436E-2(2.5E-4)†
Discontinuous	VNT3	2.5369E+0(8.7E-2)	3.8870E-2(1.4E-3)†	2.3174E+0(1.3E-1)	3.8536E-2(1.2E-3)†	2.5622E+0(8.2E-2)	3.8244E-2(1.1E-3)†
	SCH2	1.2321E-1(4.0E-2)	2.0902E-2(2.3E-4)†	1.0491E-1(2.8E-4)	2.0832E-2(2.7E-4)†	5.0208E+0(5.0E-3)	2.0861E-2(2.1E-4)†
	KUR	1.9336E-1(2.3E-2)	3.4693E-2(6.7E-4)†	4.1089E-2(2.3E-4)	3.4142E-2(6.2E-4)†	4.3389E-2(8.7E-4)	3.4960E-2(9.2E-4)†
	ZDT3	3.0837E-2(8.0E-4)	4.6028E-3(6.5E-5)†	1.0929E-2(7.9E-5)	4.7997E-3(6.0E-5)†	1.3874E-2(5.3E-3)	5.4836E-3(1.5E-4)†
Complex PS	WFG2	7.0639E-2(8.6E-3)	2.5660E-2(1.1E-2)†	3.9081E-2(3.7E-3)	2.0238E-2(4.1E-3)†	1.1083E-1(1.1E-2)	3.0526E-2(6.9E-3)†
	DTLZ7	4.2838E-1(2.9E-1)	2.7161E-1(2.5E-1)†	1.2946E-1(9.6E-4)	5.6166E-2(1.1E-3)†	1.2840E-1(9.3E-4)	5.8486E-2(9.7E-4)†
	UF1	4.6600E-2(3.4E-3)	3.5263E-2(5.4E-3)†	2.1201E-2(3.3E-2)	1.6430E-3(1.6E-4)†	7.4430E-2(6.9E-2)	9.3327E-3(2.1E-2)†
	UF2	4.6187E-2(1.9E-3)	2.2549E-2(2.8E-3)†	1.4578E-2(2.2E-2)	6.5645E-3(3.1E-3)†	5.2665E-2(5.2E-2)	1.1159E-2(2.7E-3)†
	UF3	7.1289E-2(2.8E-2)	4.2563E-2(2.3E-2)†	1.1098E-2(1.3E-2)	9.5736E-3(1.0E-2)†	3.8405E-2(3.1E-2)	1.3594E-2(1.5E-2)†
	UF4	5.2290E-2(2.0E-3)	5.0860E-2(2.6E-3)†	6.4407E-2(4.0E-3)	6.6063E-2(5.7E-3)	6.7517E-2(5.7E-3)	6.6512E-2(5.9E-3)
	UF5	1.1661E+0(1.2E-1)	5.1121E-1(1.4E-1)†	4.9510E-1(1.7E-1)	4.0341E-1(1.4E-1)†	4.6553E-1(1.3E-1)	4.7409E-1(1.5E-1)
	UF6	3.5670E-1(2.1E-2)	2.8400E-1(1.1E-2)†	5.0766E-1(1.5E-1)	4.2500E-1(1.4E-1)†	5.2782E-1(1.4E-1)	4.4913E-1(1.5E-1)†
	UF7	2.3288E-2(1.9E-3)	1.5350E-2(1.1E-3)†	2.0330E-2(1.5E-2)	1.2116E-2(5.0E-3)†	4.8084E-2(1.2E-1)	1.4598E-2(3.8E-3)†
	UF8	3.8577E-1(9.3E-3)	1.3722E-1(3.5E-2)†	5.6104E-2(2.1E-2)	1.1147E-1(4.6E-2)	8.9914E-2(3.3E-2)†	
Many Objectives	UF9	9.9491E-2(3.0E-3)	1.1081E-1(4.1E-2)	1.1732E-1(4.8E-2)	1.3153E-1(3.4E-2)†	1.3193E-1(3.7E-2)	1.2588E-1(4.1E-2)
	UF10	2.2497E+0(2.8E-1)	2.3864E+0(1.8E-1)†	4.6931E-1(6.6E-2)	4.6496E-1(7.0E-2)	4.8214E-1(1.1E-1)	4.2631E-1(6.1E-2)
DTLZ(2,6,10,20,30)	DTLZ2(4)	1.7281E-1(1.1E-3)	9.8155E-2(4.8E-3)†	8.9093E-2(7.0E-4)	9.3172E-2(3.2E-3)†	8.7399E-2(5.1E-6)	9.3679E-2(4.6E-3)†
	DTLZ2(6)	3.7149E-1(2.4E-3)	2.3931E-1(2.7E-3)†	2.8757E-1(1.4E-2)	2.3766E-1(1.9E-3)†	2.5665E-1(2.4E-5)	2.4064E-1(7.6E-4)†
	DTLZ2(10)	8.6901E-1(3.4E-2)	4.8722E-1(4.8E-3)†	5.6890E-1(1.9E-2)	4.6966E-1(7.4E-3)†	4.9222E-1(2.4E-5)	4.7015E-1(2.7E-3)†
	DTLZ5(2,10)	7.4584E-1(0.0E+0)	2.1183E-3(3.2E-5)†	1.6977E-1(1.8E-3)	2.1620E-3(7.2E-5)†	6.4940E-2(5.3E-5)	2.1492E-3(6.8E-5)†
	DTLZ5(3,10)	6.1236E-1(7.6E-2)	3.7996E-2(3.1E-3)†	2.5109E-1(3.2E-4)	3.7136E-2(1.1E-3)†	1.6867E-1(3.1E-3)	3.8918E-2(2.1E-3)†

“†” indicates that the value of the BCE algorithm is significantly different from that of its corresponding non-Pareto algorithm at a 0.05 level by the Wilcoxon’s rank sum test.

probability of parent individuals selected from the neighborhood, and the maximum number of replaced individuals in MOEA/D were specified as 10% of the population size, 0.9, and 1% of the population size, respectively. As suggested in [75] and [82], the penalty parameter θ in MOEA/D+PBI was set to 5 and the scaling factor κ in IBEA to 0.05. In BCE, the embedded non-Pareto algorithms used the same setting of parameters as in their original versions.

All the considered algorithms were given real-valued variables. The two widely used crossover and mutation operators simulated binary crossover (SBX) and polynomial mutation (with distribution indexes 20 [15]) were used on all the MOPs, except UF. The crossover probability was set to $p_c = 1.0$ and mutation probability to $p_m = 1/d$, where d denotes the number of decision variables. For the UF problems that have a strong linkage in variables, the use of variable-independent SBX may not be adequate [16], [44]. Following the study in [44] and [74], we adopted the DE operation for these problems, with the two control parameters $CR = 1.0$ and $F = 0.5$.

Tables III and IV give the HV and IGD results (mean and standard deviation), respectively, for the three groups of

paired algorithms, IBEA versus BCE-IBEA, MOEA/D+TCH versus BCE-MOEA/D+TCH, and MOEA/D+PBI versus BCE-MOEA/D+PBI, on all 42 MOPs. The better mean for each problem is highlighted in boldface. To have statistically sound conclusions, the Wilcoxon’s rank sum test [81] at a 0.05 significance level is adopted to test the significance of the differences between the results obtained by paired algorithms.

As stated before, the advantage of NPC is primarily on addressing challenging MOPs (such as with a complex PS or with a high-dimensional objective space), whereas the advantage of PC lies in dealing with MOPs with an irregular Pareto front. Here, we divide the test problems into seven categories to systematically investigate the effectiveness of BCE for problems with distinct preference of NPC or PC. The categories are convex, concave, linear, mixed, discontinuous, complex-PS, and high-dimensional problem.

A. Test Problems With Convex Pareto Front

In this category, we consider four problems, such as SCH1, ZDT1, ZDT4, and VNT1. As can be seen in Tables III and IV, the BCE algorithms show a clear advantage over the non-Pareto algorithms on these problems. The three

TABLE IV
HV RESULTS (MEAN AND SD) OF THE THREE GROUPS OF PAIRED ALGORITHMS. THE BETTER MEAN FOR EACH CASE IS HIGHLIGHTED IN BOLDFACE

Property	Problem	IBEA	BCE-IBEA	MOEA/D+TCH	BCE-MOEA/D+TCH	MOEA/D+PBI	BCE-MOEA/D+PBI
Convex	SCH1	1.0395E+0(8.0E-5)	1.0396E+0(5.0E-5)†	1.0375E+0(1.7E-4)	1.0396E+0(5.4E-5)†	1.0275E+0(5.6E-4)	1.0396E+0(3.5E-5)†
	ZDT1	8.7140E-1(1.2E-4)	8.7156E-1(1.0E-4)†	8.6969E-1(5.5E-4)	8.6998E-1(3.1E-4)†	8.6149E-1(2.6E-3)	8.6639E-1(8.4E-4)†
	ZDT4	3.5648E-1(1.1E-1)	7.8067E-1(4.1E-2)†	8.5850E-1(6.1E-3)	8.6174E-1(5.8E-3)†	8.1319E-1(4.4E-2)	8.5370E-1(9.8E-3)†
	VNT1	9.8473E-1(1.2E-2)	1.0134E+0(1.2E-3)†	9.7156E-1(2.5E-4)	1.0144E+0(1.3E-3)†	9.9419E-1(3.1E-4)	1.0138E+0(9.2E-4)†
Concave	ZDT2	5.3694E-1(1.1E-4)	5.3793E-3(1.0E-4)†	5.3587E-1(6.3E-4)	5.3654E-1(4.1E-4)†	5.2098E-1(2.6E-3)	5.2856E-1(1.8E-3)†
	ZDT6	6.0575E-1(4.7E-4)	6.0900E-1(3.4E-4)†	5.8120E-1(3.7E-2)	5.9186E-1(2.8E-3)†	5.2924E-1(9.4E-3)	5.6742E-1(5.6E-3)†
	WFG4	4.1692E-1(4.6E-4)	4.1786E-1(4.9E-4)†	4.1132E-1(1.3E-3)	4.1254E-1(9.7E-4)†	3.9106E-1(3.5E-3)	3.9880E-1(2.1E-3)†
	WFG5	3.7429E-1(1.6E-3)	3.7556E-1(1.8E-3)†	3.7089E-1(3.6E-4)	3.7175E-1(6.7E-4)†	3.6200E-1(1.9E-3)	3.6589E-1(1.1E-3)†
	WFG6	3.8234E-1(6.4E-3)	3.8235E-1(6.1E-3)	3.7690E-1(7.1E-3)	3.8085E-1(5.3E-3)†	3.5462E-1(8.8E-3)	3.6529E-1(7.8E-3)†
	WFG7	4.1791E-1(2.4E-4)	4.1933E-1(2.4E-4)†	4.1642E-1(7.6E-4)	4.1729E-1(4.8E-4)†	3.9833E-1(3.1E-3)	4.0522E-1(1.6E-3)†
	WFG8	3.5056E-1(3.5E-3)	3.5539E-1(3.3E-3)†	3.5389E-1(3.2E-3)	3.5561E-1(3.2E-3)†	3.3363E-1(4.9E-3)	3.4172E-1(5.1E-3)†
	WFG9	3.6836E-1(3.4E-2)	3.6929E-1(3.4E-2)†	3.9231E-1(1.5E-2)	3.8015E-1(2.8E-2)†	3.7718E-1(1.1E-2)	3.6887E-1(2.4E-2)
	DTLZ2	7.4369E-1(5.5E-4)	7.4216E-1(2.0E-3)†	7.4893E-1(1.3E-4)	7.4760E-1(1.1E-3)†	7.4897E-1(7.9E-5)	7.4745E-1(1.1E-3)†
	DTLZ3	2.4094E-1(8.1E-2)	2.4504E-1(1.7E-1)	3.8094E-1(2.4E-1)	9.4924E-2(1.8E-1)†	2.2132E-1(2.7E-1)	1.0137E-1(1.8E-1)
	DTLZ4	5.1710E-1(2.2E-1)	6.4365E-1(2.0E-1)†	6.8054E-1(1.2E-1)	7.4702E-1(1.3E-3)†	7.0945E-1(1.0E-1)	7.4745E-1(1.3E-3)†
	DTLZ5	2.6296E-1(2.5E-4)	2.6596E-1(2.7E-4)†	2.5645E-1(8.0E-6)	2.6593E-1(3.1E-4)†	2.4415E-1(7.1E-5)	2.6576E-1(1.8E-4)†
	DTLZ6	1.5127E-1(3.9E-2)	1.7524E-1(3.3E-2)†	3.0630E-0(3.7E-3)†	0.0000E+0(0.0E+0)	0.0000E+0(0.0E+0)	0.0000E+0(0.0E+0)
Linear	WFG3	6.9936E-1(1.3E-3)	6.9938E-1(1.5E-3)	6.9099E-1(3.1E-3)	6.9282E-1(2.5E-3)†	6.7584E-1(5.6E-3)	6.8170E-1(4.8E-3)†
	DTLZ1	5.0330E-1(9.1E-2)	1.1126E+0(8.1E-3)†	1.1193E+0(4.9E-3)	1.1150E+0(8.0E-3)†	1.1194E+0(3.3E-3)	1.1025E+0(3.7E-2)†
Mixed	WFG1	3.5198E-1(4.0E-2)	3.4025E-1(4.2E-2)	2.1352E-1(3.8E-2)	2.5791E-1(3.9E-2)†	1.5041E-1(2.3E-2)	2.2037E-1(3.8E-2)†
	VNT2	1.2258E+0(9.5E-3)	1.2481E+0(4.4E-4)†	1.2179E+0(1.2E-4)	1.2483E+0(3.4E-4)†	1.2053E+0(4.8E-4)	1.2479E+0(3.4E-4)†
	VNT3	1.1427E+0(4.5E-3)	1.1476E+0(3.7E-4)†	1.1255E+0(9.1E-4)	1.4756E+0(3.2E-4)†	1.1296E+0(5.0E-4)	1.1476E+0(4.1E-4)†
Discontinuous	SCH2	8.0570E-1(2.4E-3)	8.1170E-1(6.4E-5)†	8.0177E-1(6.0E-5)	8.1168E-1(1.6E-4)†	6.4919E-1(3.6E-5)	8.1163E-1(4.0E-4)†
	KUR	6.0476E-1(6.8E-4)	6.1110E-1(1.5E-4)†	6.0987E-1(1.8E-4)	6.1108E-1(1.9E-4)†	6.0806E-1(4.1E-4)	6.1044E-1(3.1E-4)†
	ZDT3	7.2021E-1(4.9E-4)	7.2542E-1(3.1E-4)†	7.2236E-1(2.9E-4)	7.2448E-1(2.4E-4)†	7.1218E-1(4.3E-3)	7.2140E-1(4.9E-4)†
	WFG2	7.3840E-1(8.1E-3)	7.3850E-1(1.4E-2)	7.4146E-1(1.2E-2)	7.4707E-1(1.2E-2)†	7.1395E-1(8.5E-3)	7.3599E-1(1.1E-2)†
	DTLZ7	4.5444E-1(8.3E-2)	5.0576E-1(6.2E-2)†	5.3340E-1(4.1E-4)	5.6217E-1(1.6E-3)†	5.1590E-1(2.7E-3)	5.5925E-1(2.0E-3)†
Complex PS	UF1	8.0299E-1(4.9E-3)	8.1707E-1(9.3E-3)†	8.1707E-1(9.3E-3)	8.7067E-1(1.5E-2)†	8.1090E-1(5.4E-2)	8.6548E-1(2.0E-2)†
	UF2	8.0179E-1(3.0E-3)	8.4197E-1(3.4E-3)†	8.5721E-1(2.2E-2)	8.6395E-1(1.2E-2)†	8.2907E-1(3.5E-2)	8.6149E-1(2.9E-3)†
	UF3	7.6131E-1(4.2E-2)	8.0534E-1(3.6E-2)†	8.5673E-1(2.5E-2)	8.5955E-1(1.2E-2)†	8.0639E-1(5.3E-2)	8.3706E-1(3.9E-2)†
	UF4	4.4840E-1(4.3E-3)	4.5528E-1(5.8E-3)†	4.3154E-1(6.6E-3)	4.3115E-1(9.4E-3)	4.2700E-1(9.6E-3)	4.3010E-1(9.3E-3)
	UF5	0.0000E+0(0.0E+0)	2.9497E-2(4.3E-2)†	1.9400E-1(9.7E-2)	2.5115E-1(9.8E-2)†	1.9166E-1(1.1E-1)	1.9518E-1(1.0E-1)
	UF6	1.2521E-1(1.7E-2)	2.2287E-1(4.8E-2)†	2.6455E-1(7.3E-2)	2.9623E-1(6.0E-2)†	2.3798E-1(7.0E-2)	2.6663E-1(6.2E-2)†
	UF7	6.7217E-1(2.9E-3)	6.8245E-1(1.9E-3)†	6.7607E-1(2.1E-2)	6.8884E-1(1.5E-2)†	6.5252E-1(9.8E-2)	6.7787E-1(1.3E-2)†
	UF8	2.8638E-1(8.7E-3)	5.5758E-1(4.6E-2)†	6.8837E-1(3.2E-2)	6.7743E-1(2.8E-2)	5.6021E-1(8.9E-2)	6.0717E-1(6.3E-2)†
	UF9	9.0647E-1(8.9E-3)	8.9953E-1(5.6E-2)	9.1761E-1(7.7E-2)	8.9569E-1(5.7E-2)	8.9252E-1(5.9E-2)	9.0663E-1(6.8E-2)
	UF10	0.0000E+0(0.0E+0)	0.0000E+0(0.0E+0)	1.1111E-1(3.5E-2)	1.1990E-1(3.3E-2)	1.7454E-1(4.4E-2)	1.7519E-1(4.3E-2)
Many Objectives	DTLZ2(4)	1.0506E+0(6.0E-4)	1.0428E+0(2.7E-3)†	1.0585E+0(2.0E-4)	1.0538E+0(1.2E-3)†	1.0588E+0(3.2E-5)	1.0536E+0(1.2E-3)†
	DTLZ2(6)	1.5551E+0(9.7E-4)	1.5324E+0(3.4E-3)†	1.5232E+0(1.1E-2)	1.5311E+0(4.0E-3)†	1.5504E+0(8.5E-4)	1.5423E+0(2.4E-3)†
	DTLZ2(10)	1.9101E+0(1.3E-1)	2.5049E+0(3.8E-3)†	2.4452E+0(2.5E-2)	2.4701E+0(7.4E-3)†	2.5092E+0(3.7E-4)	2.4954E+0(3.7E-3)†
	DTLZ5(2,10)	2.3597E-1(8.8E-4)	2.6133E-1(1.2E-3)†	2.3582E-1(5.9E-4)	2.6095E-1(1.4E-3)†	2.5850E-1(7.8E-4)	2.6078E-1(1.6E-3)†
	DTLZ5(3,10)	4.1017E-1(9.7E-2)	8.7522E-1(5.3E-3)†	7.5172E-1(1.1E-3)	8.7512E-1(7.3E-3)†	7.5719E-1(8.9E-3)	8.7723E-1(6.6E-3)†

“†” indicates that the value of the BCE algorithm is significantly different from that of its corresponding non-Pareto algorithm at a 0.05 level by the Wilcoxon’s rank sum test.

BCE algorithms outperform their corresponding competitors for both IGD and HV on all four problems, and the difference in all of these comparisons is statistically significant.

Fig. 4 plots the final solutions of the six algorithms in a single run on SCH1. This particular run, along with others for visual demonstration in the paper, is associated with the result that is the closest to the mean IGD value. SCH1 has a convex Pareto optimal curve in the range $f_1, f_2 \in [0, 4]$. As shown, IBEA and MOEA/D+TCH struggle to maintain the uniformity of the solutions, especially around the edges of the Pareto front. MOEA/D+PBI fails to find boundary points of the Pareto front, with their solutions concentrating in the range $[0, 3]$. On the other hand, the three BCE algorithms perform well. Their performance appears similar, and all of their solutions are uniformly distributed along the whole Pareto front. This is attributable to the population maintenance operation in BCE, which can effectively eliminate poorly distributed solutions in the evolutionary process. In the rest of this paper, for brevity, we plot the solutions of only one of the BCE algorithms, if they perform visually similarly.

In addition, Fig. 5 shows the final solutions on VNT1. Clearly, for this three-objective problem, only the BCE

algorithms have good diversity. The solutions obtained by IBEA are solely located in the middle of the Pareto front. The solutions of the two MOEA/D algorithms, which correspond to uniformly distributed weight vectors, exhibit a specific structure but do not have a good distribution over the desired front.

B. Test Problems With Concave Pareto Front

In this category, we consider 13 problems from the ZDT, WFG, and DTLZ problem suites. As can be seen in Tables III and IV, the three BCE algorithms generally perform better than their competitors. Specifically, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI obtain a better IGD value in 12, 8, and 9 out of the 13 test instances, respectively. For HV, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI outperform their corresponding non-Pareto algorithms in 12, 9, and 9 out of the 13 instances, respectively.

In fact, for some MOPs (such as DTLZ2), some non-Pareto algorithms already work quite well. In this case, the exploration operation in the PC evolution of BCE can hardly further

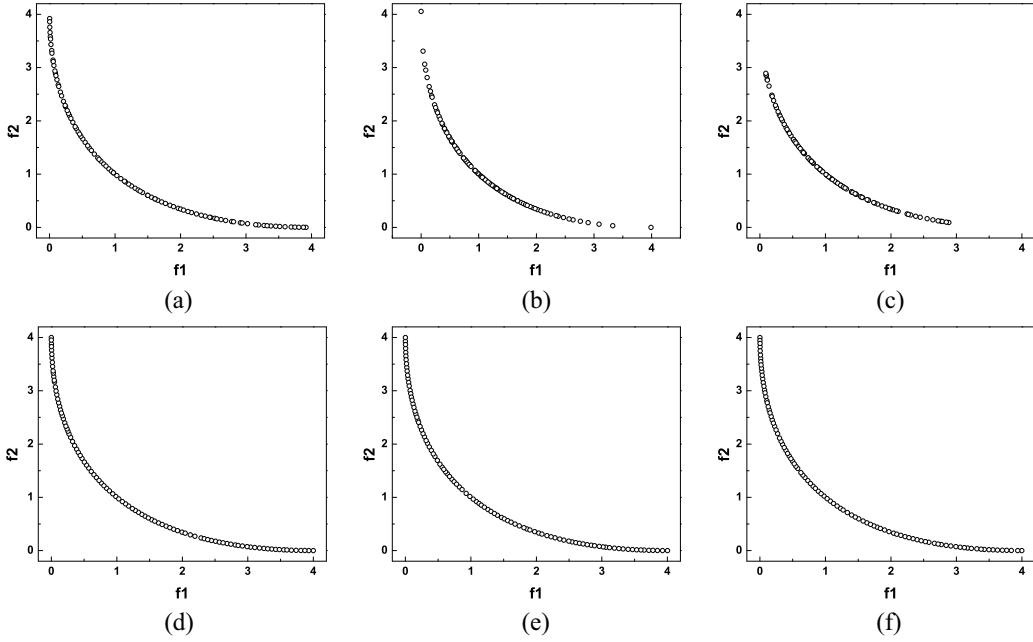


Fig. 4. Final solution set of the six algorithms on SCH1. (a) IBEA. (b) MOEA/D+TCH. (c) MOEA/D+PBI. (d) BCE-IBEA. (e) BCE-MOEA/D+TCH. (f) BCE-MOEA/D+PBI.

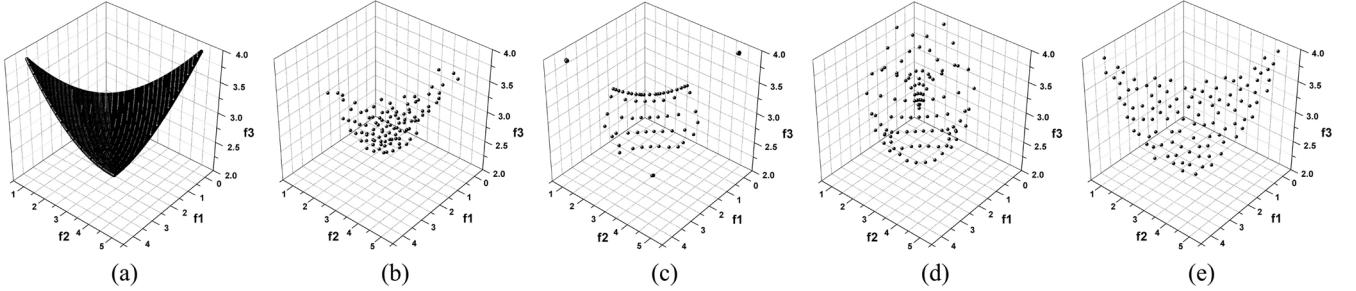


Fig. 5. Pareto front and the final solution set on VNT1, where the solutions of the three BCE algorithms have similar distribution. (a) Pareto front. (b) IBEA. (c) MOEA/D+TCH. (d) MOEA/D+PBI. (e) BCE-MOEA/D+TCH.

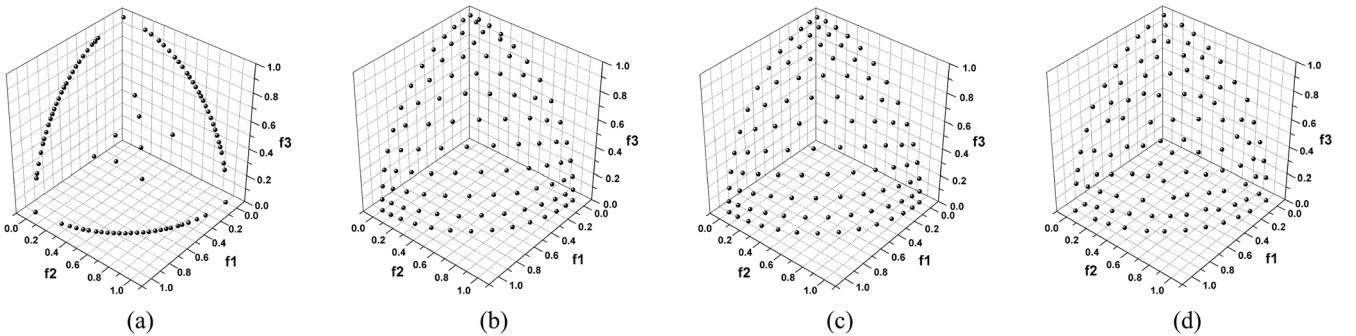
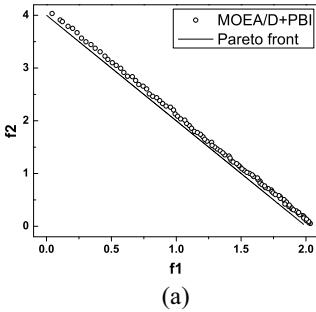


Fig. 6. Final solution set on DTLZ2, where the solutions of the three BCE algorithms have similar distribution. (a) IBEA. (b) MOEA/D+TCH. (c) MOEA/D+PBI. (d) BCE-MOEA/D+TCH.

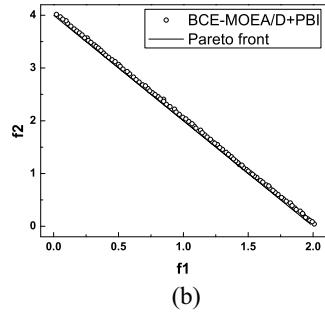
improve individuals' performance but can lead to the decrease in the computational resources (i.e., function evaluations) occupied by the NPC evolution. Fig. 6 gives the final solutions obtained by IBEA, MOEA/D+TCH, MOEA/D+PBI, and BCE-MOEA/D+TCH on DTLZ2. Clearly, for this problem, IBEA is unable to maintain uniformity of the solutions, but MOEA/D+TCH and MOEA/D+PBI have a set of excellently distributed solutions over the Pareto front. This is

consistent with the result in Table III, where IBEA performs worse than BCE-IBEA, but the two MOEA/D algorithms perform better than their competitors.

As to the statistical results, it can be observed from the tables that the difference between the paired algorithms is significant for most of the test instances. Specifically, the proportion of the test instances where the three BCE algorithms BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI



(a)



(b)

Fig. 7. Final solution set obtained by (a) MOEA/D+PBI and (b) BCE-MOEA/D+PBI on WFG3.

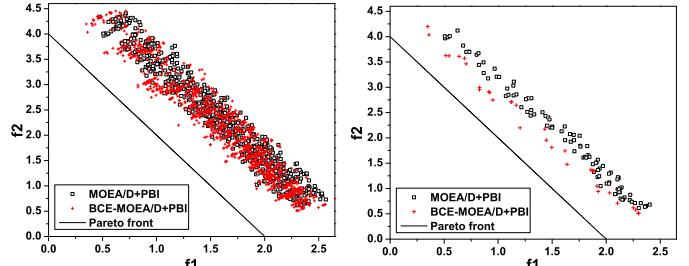
outperform their competitors with statistical significance is 11/13, 6/13, and 9/13 for IGD and 9/13, 9/13, and 9/13 for HV, respectively. Conversely, the proportion of the instances where the three non-Pareto algorithms IBEA, MOEA/D+TCH, and MOEA/D+PBI are superior with statistical significance is 0/13, 4/13, and 3/13 for IGD and 1/13, 4/13, and 1/13 for HV, respectively.

C. Test Problems With Linear Pareto Front

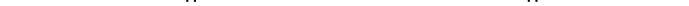
Non-Pareto EMO algorithms in general work well on this kind of problem as their NPC is not likely to prefer specific areas of a plane Pareto front. Despite that, the proposed approach is still competitive, as can be seen from the results on test problems WFG3 and DTLZ1 in Tables III and IV. For WFG3, the three BCE algorithms all outperform their competitors. For a visual comparison, Fig. 7 plots the final solutions of MOEA/D+PBI and BCE-MOEA/D+PBI as well as the problem's Pareto front. As shown, BCE-MOEA/D+PBI has a better performance than MOEA/D+PBI in terms of both diversity and convergence. This observation is interesting because it is commonly believed that the solutions guided by an NPC have a better convergence than those by the PC. One important reason for this occurrence is that the exploration around the nondominated solutions in BCE can effectively drive the population evolving toward the Pareto front, especially at the initial stage of evolution.

To take a closer look, Fig. 8 gives the results of MOEA/D+PBI and BCE-MOEA/D+PBI during the initial 1000 function evaluations, where Fig. 8(a) plots all 1000 individuals produced by the two algorithms and Fig. 8(b) plots their evolutionary population at the 1000 evaluations. As can be seen in Fig. 8(a), there exist some individuals of BCE-MOEA/D+PBI apparently closer to the optimal front. This is the result of effective exploration of the nondominated individuals (i.e., the PC population) in BCE. These nondominated individuals, whose number is smaller than the population capacity at that time, can represent the best individuals found so far, as seen from the comparison between Fig. 8(a) and (b).

The test function DTLZ1 has a huge number of local optimal fronts ($11^5 - 1$). For this problem, BCE-IBEA outperforms IBEA, but BCE-MOEA/D+TCH and BCE-MOEA/D+PBI perform worse than the two MOEA/D algorithms. In Section V-C, we will provide a detailed explanation for why BCE may be outperformed by some non-Pareto algorithms on such MOPs with a number of local optima.



(a)



(b)

Fig. 8. Results of MOEA/D+PBI and BCE-MOEA/D+PBI during initial 1000 function evaluations on WFG3. (a) All the individuals produced during the 1000 evaluations. (b) Evolutionary population at the 1000 evaluations.

D. Test Problems With Mixed Pareto Front

The results of three of this kind of problems, WFG1, VNT2 and VNT3, are shown in Tables III and IV, where the BCE algorithms significantly outperform their competitors. They are superior with statistical significance in eight out of the nine comparisons for both IGD and HV indicators. For a visual observation, Fig. 9 plots the final solutions obtained by IBEA, MOEA/D+TCH, MOEA/D+PBI, and BCE-MOEA/D+TCH on VNT3. Clearly, only BCE-MOEA/D+TCH has well-distributed solutions over the whole Pareto front. The solutions obtained by the three non-Pareto algorithms concentrate mainly in the middle segment and fail to extend to the left part of the optimal front.

E. Test Problems With Discontinuous Pareto Front

As can be seen from the two tables, for MOPs with a discontinuous Pareto front, the proposed approaches have a clear advantage over the non-Pareto algorithms. The three BCE algorithms significantly outperform their competitors for all the instances, and on most of these instances they even have an order of magnitude smaller IGD values.

In fact, non-Pareto algorithms commonly struggle to maintain the diversity of solutions on this kind of MOPs. This happens mainly because the imaginary parts of the discontinuous Pareto front largely affect the accuracy of the fitness estimation based on an NPC. For example, in MOEA/D, the breakpoints of the discontinuous Pareto front may correspond to the optimal solution of multiple scalar subproblems [58]. This is likely to cause the failure of uniformity maintenance of solutions, further leading to the search of the algorithm only on some specific regions of the objective space. Fig. 10 plots the final solutions obtained by IBEA, MOEA/D+TCH, MOEA/D+PBI, and BCE-MOEA/D+TCH on SCH2. It can be observed that IBEA and MOEA/D+TCH are unable to maintain the uniformity of solutions, and MOEA/D+PBI fails to find the upper part of the Pareto front. Note that there exist some dominated solutions in the set of solutions obtained by MOEA/D+PBI because a dominated solution may have a closer distance than a nondominated one to the corresponding reference point in MOEA/D+PBI, as pointed out in [16].

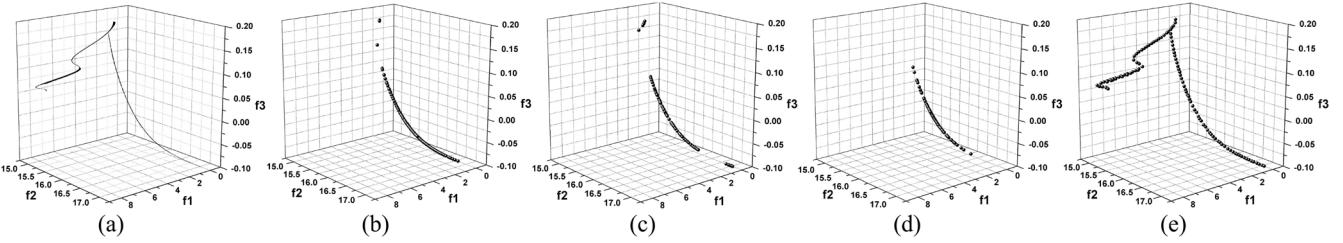


Fig. 9. Pareto front and the final solution set on VNT3 in which the solutions of the three BCE algorithms have similar distribution. (a) Pareto front. (b) IBEA. (c) MOEA/D+TCH. (d) MOEA/D+PBI. (e) BCE-MOEA/D+TCH.

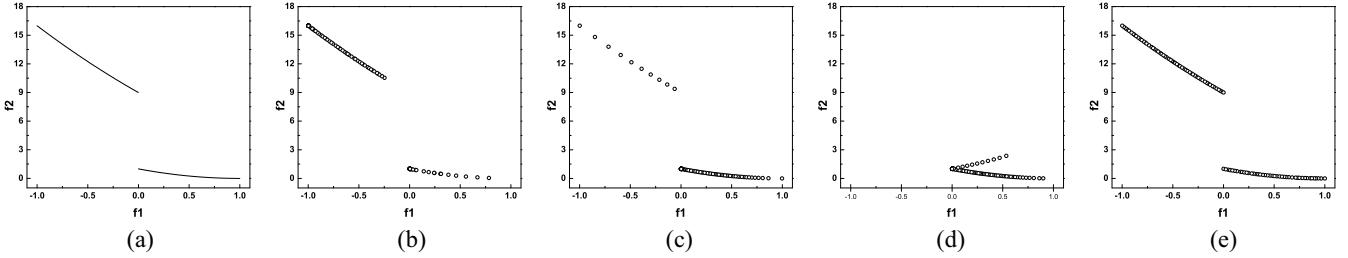


Fig. 10. Pareto front and the final solution set on SCH2 in which the solutions of the three BCE algorithms have similar distribution. (a) Pareto front. (b) IBEA. (c) MOEA/D+TCH. (d) MOEA/D+PBI. (e) BCE-MOEA/D+TCH.

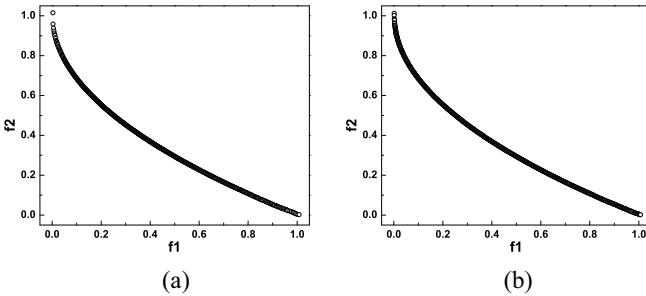


Fig. 11. Final solution set obtained by (a) MOEA/D+TCH and (b) BCE-MOEA/D+TCH on UF1.

F. Test Problems With Complex PS

In this section, we consider the UF problem suite from the CEC2009 competition [74]. These MOPs involve a strong linkage in variables among the Pareto optimal solutions, thereby posing a big challenge for EMO algorithms [44], [76]. In spite of that, the BCE algorithms outperform their corresponding non-Pareto algorithms on the majority of the test instances, as shown in Tables III and IV. Specifically, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI achieve a better IGD value than their competitors in eight, seven, and nine out of the ten test instances, respectively. For HV, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI outperform their competitors in eight, seven, and ten out of the ten instances, respectively. Also, the difference in most of these comparisons is statistically significant, with the winning ratio of the BCE algorithms against their competitors being 19 to 2 for IGD and 19 to 0 for HV in the 30 comparisons, respectively. One reason for this occurrence is likely due to the population maintenance operation in BCE, which is able to maintain the diversity of solutions effectively.

Fig. 11 plots the final solutions obtained by MOEA/D+TCH and BCE-MOEA/D+TCH on UF1. As shown, although a good distribution of solutions is obtained in most parts of the Pareto front by MOEA/D+TCH, there is a clear interval between the upper bound and the other solutions. In contrast, the solutions obtained by BCE-MOEA/D+TCH have a good distribution uniformity along the whole front.

G. Test Problems With Many Objectives

In this section, test problems DTLZ2 [18] and DTLZ5(I, m) [62], [64] are used to verify the performance of BCE on many-objective problems. DTLZ2 has a spherical Pareto front in the range $f_1, f_2, \dots, f_m \in [0, 1]$, and DTLZ5(I, m) has a degenerate Pareto front, with its dimensionality I lower than that of the objective space m .

Tables III and IV show the results of the six algorithms on five instances of DTLZ2 and DTLZ5(I, m). These instances are three DTLZ2 functions with four, six, and ten objectives, and two 10-objective DTLZ5(I, m) functions with 2-D and 3-D Pareto front, i.e., DTLZ5(2,10) and DTLZ5(3,10). As can be seen from the tables, the compared algorithms perform similarly on the four- and six-objective instances in which the BCE algorithms often have a better IGD result, whereas the non-Pareto algorithms generally obtain a higher HV value. For three 10-objective instances, the BCE algorithms significantly outperform their competitors. This suggests that the advantage of the proposed approach becomes clearer when a higher-dimensional space is involved. Fig. 12 shows the final solutions of MOEA/D+TCH and BCE-MOEA/D+TCH on the ten-objective DTLZ2 by parallel coordinates. As shown, MOEA/D+TCH fails to find widely distributed solutions on objectives f_1 to f_3 , which is in contrast to the result of BCE-MOEA/D+TCH whereby a spread of solutions over $f_i \in [0, 1]$ is obtained.

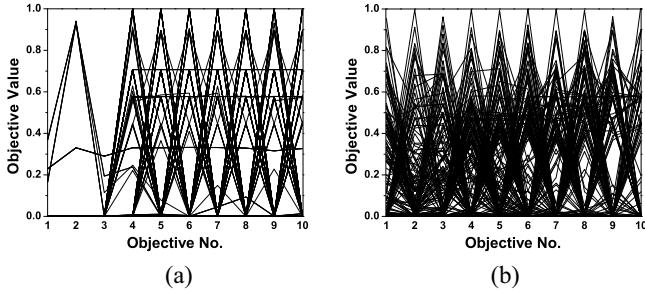


Fig. 12. Final solution set obtained by (a) MOEA/D+TCH and (b) BCE-MOEA/D+TCH on the ten-objective DTLZ2, as shown by parallel coordinates.

In DTLZ5(I, m), all objectives within $\{f_1, \dots, f_{m-I+1}\}$ are positively correlated, whereas the objectives in $\{f_{m-I+1}, \dots, f_m\}$ conflict with each other. Fig. 13 plots the final solutions of IBEA, MOEA/D+TCH, MOEA/D+PBI, and BCE-MOEA/D+TCH in the last three objectives (f_8, f_9, f_{10}) of DTLZ5(3,10). The Pareto optimal solutions of the problem with respect to these objectives satisfy $2f_8^2 + f_9^2 + f_{10}^2 = 1$. As shown, despite several solutions of BCE-MOEA/D+TCH not located on the optimal front, the rest has good uniformity and coverage over the whole front. In contrast, the three non-Pareto algorithms struggle to maintain diversity, with their solutions concentrating in some tiny parts (or even several points) of the Pareto front.

Finally, it is worth mentioning that Pareto-based algorithms are often seen as failing to deal with many-objective problems. Their Pareto dominance and density-based selection criteria even could push the population against the optimal front in a high-dimensional space [31], [47], [68]. Interestingly, the solution set of BCE, which is the PC population maintained by Pareto dominance and density, performs well in many-objective problems. This occurrence can be attributed to the role of the NPC evolution in BCE, which leads the PC population to evolve toward the desired direction.

H. Result Summary

To summarize, the BCE algorithms generally outperform their corresponding non-Pareto algorithms. BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI have a better IGD value in 38, 32, and 35 out of the 42 test problems. For HV, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI perform better in 36, 33, and 34 out of the 42 test problems. Note that for a couple of test problems, some paired algorithms obtain different comparison results with respect to IGD and HV, although both indicators involve comprehensive performance of convergence and diversity. For example, for the four- and six-objective DTLZ2, BCE-IBEA has a better IGD but worse HV than the original IBEA. This contradiction between HV and IGD happens more on the problems with a concave Pareto front, as reported in [38]. The reason for this occurrence is the different preference of the two indicators [50]. IGD, which is based on uniformly distributed points along the entire Pareto front, prefers the distribution uniformity of the solution set, whereas HV, which is typically

influenced more by the boundary solutions, has a bias toward the extensivity of the solution set.

V. FURTHER INVESTIGATIONS OF BCE

Having demonstrated its competitiveness on various test problems above, BCE is further investigated in this section for a deeper understanding of its behavior. Due to the space limit and similar comparative results obtained by the IGD and HV indicators, we present only the IGD results in this and the following sections.

A. Performance Verification of the NPC Evolution

The previous experimental results have shown the effectiveness of the PC evolution in maintaining the individual diversity and approaching the optimal front. A question may now arise regarding the NPC evolution. Does the NPC evolution benefit from the information exchange with the PC evolution? In other words, can non-Pareto EMO algorithms themselves benefit when working under this BCE framework?

To answer this question, we give the IGD results between the solution set of the original non-Pareto algorithms and that of the embedded ones (i.e., the NPC population) on the 42 test problems in Table V. As shown, for most of the problems, the performance of the three non-Pareto algorithms is improved when working under the BCE framework. The NPC population of BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI has a better IGD value than the solution set of the corresponding non-Pareto algorithms in 31, 31, and 32 out of all the 42 instances, respectively.

In addition, it is worth mentioning that unlike the PC evolution that can preserve any nondominated individual located in a sparse region, the NPC evolution may not preserve such individuals due to its own selection criterion. That is, even when the PC evolution produces plenty of promising individuals that clearly help enhance the population diversity, they may still not enter the NPC population if not preferred by the considered NPC. That is why, for some MOPs in which the PC population significantly outperforms that of the non-Pareto algorithm, the NPC population yet performs similarly to (or even slightly worse than) the latter, such as the results of BCE-MOEA/D+TCH on DTLZ5(2,10). Interestingly, there are also some exceptions that the population of non-Pareto algorithms has a clear improvement when working collaboratively with the PC population in BCE. Fig. 14 gives such an example, where the solution set of the original IBEA and the one working under the BCE framework (i.e., the NPC population of BCE-IBEA) are plotted by parallel coordinates. Clearly, in contrast to IBEA that fails to find diverse solutions on objectives f_1 to f_6 , the NPC evolution in BCE-IBEA maintains a good diversity of population for all the objectives of the Pareto front.

In summary, despite evolving based on their own selection criterion, the non-Pareto algorithms generally show a performance improvement when embedded into the NPC evolution part of BCE. This, along with the experimental results in the previous section, indicates that both the NPC and PC

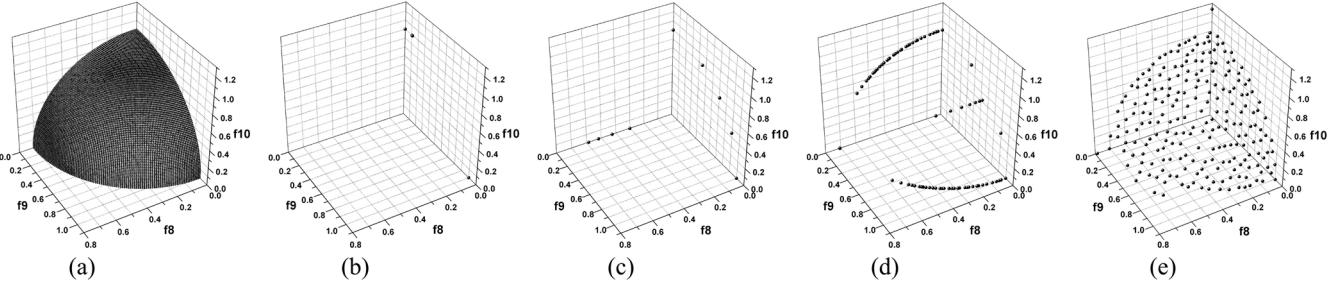


Fig. 13. Pareto front and the final solution set in the subspace (f_8-f_{10}) of DTLZ5(3, 10), where the solutions of the three BCE algorithms have similar distribution. (a) Pareto front. (b) IBEA. (c) MOEA/D+TCH. (d) MOEA/D+PBI. (e) BCE-MOEA/D+TCH.

TABLE V
IGD COMPARISON BETWEEN THE ORIGINAL NON-PARETO ALGORITHMS AND THE NPC EVOLUTION IN THEIR CORRESPONDING BCE ALGORITHMS. THE BETTER MEAN FOR EACH CASE IS HIGHLIGHTED IN BOLDFACE

Problem	IBEA	BCE-IBEA	MOEA/D+TCH	BCE-MOEA/D+TCH	MOEA/D+PBI	BCE-MOEA/D+PBI
SCH1	1.9093E-2(4.1E-4)	2.0162E-2(1.1E-3)	4.9139E-2(1.5E-3)	4.6963E-2(7.7E-4)	1.7765E-1(5.2E-3)	1.7910E-1(2.0E-3)
SCH2	1.2321E-1(4.0E-2)	5.7079E-2(2.0E-2)	1.0491E-1(2.8E-4)	1.0488E-1(1.1E-4)	5.0208E+0(5.0E-3)	5.0155E+0(8.7E-4)
KUR	1.9336E-1(2.3E-2)	1.6513E-1(3.0E-2)	4.1089E-2(2.3E-4)	4.1073E-2(3.3E-4)	4.3389E-2(8.7E-4)	4.2867E-2(9.1E-4)
ZDT1	4.0773E-3(6.8E-5)	4.1000E-3(6.7E-5)	4.3626E-3(3.4E-4)	4.1924E-3(1.1E-4)	8.2654E-3(1.7E-3)	6.6341E-3(5.8E-4)
ZDT2	9.1909E-3(3.7E-4)	9.0121E-1(9.8E-4)	4.3342E-3(1.8E-4)	4.1933E-3(1.2E-4)	1.0540E-2(1.2E-3)	8.2292E-3(1.0E-3)
ZDT3	3.0837E-2(8.0E-4)	3.0412E-2(1.6E-3)	1.0929E-2(7.9E-5)	1.0923E-2(1.1E-4)	1.3874E-2(5.3E-3)	1.1675E-2(4.5E-4)
ZDT4	6.2464E-1(1.1E-1)	1.3199E-1(9.5E-4)	1.0001E-2(3.4E-3)	8.9033E-3(3.3E-3)	3.7084E-2(2.8E-2)	1.4819E-2(6.7E-3)
ZDT6	5.4849E-3(1.8E-4)	5.5631E-3(1.6E-4)	1.5027E-2(1.7E-3)	1.0515E-2(1.3E-3)	3.9299E-2(4.6E-3)	2.3791E-2(2.9E-3)
WFG1	7.5883E-1(6.2E-2)	7.8579E-1(6.0E-2)	1.0487E+0(7.2E-2)	9.6474E-1(5.3E-2)	1.1983E+0(5.9E-2)	1.0520E+0(5.2E-2)
WFG2	7.0639E-2(8.6E-3)	6.9895E-2(8.5E-3)	3.9081E-2(3.7E-3)	3.8934E-2(3.8E-3)	1.1083E-1(1.1E-2)	9.3369E-2(6.8E-3)
WFG3	1.5944E-2(1.2E-3)	1.7071E-2(2.7E-3)	2.5079E-2(4.3E-3)	2.3297E-2(3.2E-3)	4.7442E-2(8.1E-3)	4.2171E-2(8.1E-3)
WFG4	1.8370E-2(8.3E-4)	1.8880E-2(1.0E-3)	2.2551E-2(1.8E-3)	2.1534E-2(1.5E-3)	5.0721E-2(5.2E-3)	4.5001E-2(4.4E-3)
WFG5	7.2029E-2(7.6E-4)	7.1733E-2(9.4E-4)	6.8459E-2(5.9E-4)	6.8496E-2(5.2E-4)	8.2498E-2(2.9E-3)	7.9373E-2(2.4E-3)
WFG6	6.5849E-2(9.4E-3)	6.5980E-2(7.9E-3)	6.8500E-2(1.0E-2)	6.3961E-2(8.1E-3)	1.0092E-1(1.4E-2)	9.0327E-2(1.2E-2)
WFG7	2.1525E-2(1.0E-3)	2.1004E-2(9.9E-4)	1.6770E-2(4.7E-4)	1.6614E-2(2.5E-4)	3.7482E-2(3.3E-3)	3.3662E-2(2.4E-3)
WFG8	9.3529E-2(8.0E-3)	3.3918E-2(2.4E-3)	8.7532E-2(5.6E-3)	8.5901E-2(5.8E-3)	1.1848E-1(7.9E-3)	1.1040E-1(8.9E-3)
WFG9	7.8794E-2(5.2E-2)	7.8164E-2(5.2E-2)	3.5633E-2(2.4E-2)	5.7653E-2(4.4E-2)	5.6432E-2(1.6E-2)	7.6906E-2(3.7E-2)
VNT1	1.6688E-1(2.7E-2)	1.3937E-1(6.9E-3)	2.2398E-1(1.3E-3)	2.2369E-1(1.1E-3)	1.7364E-1(5.6E-4)	1.7361E-1(6.4E-4)
VNT2	4.2602E-2(9.4E-3)	2.7724E-2(8.5E-3)	4.6425E-2(2.9E-4)	4.6404E-2(3.4E-4)	5.8432E-2(2.4E-4)	5.8373E-2(3.4E-4)
VNT3	2.5369E+0(8.7E-2)	2.2216E+0(4.4E-1)	2.3174E+0(1.3E-1)	2.2537E+0(1.0E-1)	2.5622E+0(8.2E-2)	2.4384E+0(5.5E-3)
DTLZ1	1.8267E-1(1.7E-2)	1.0071E-1(2.3E-2)	1.9571E-2(1.1E-3)	1.9465E-2(6.8E-4)	1.9130E-2(5.7E-4)	1.9507E-2(1.1E-3)
DTLZ2	1.1894E-1(2.2E-3)	1.1820E-1(2.5E-3)	5.1151E-2(4.2E-4)	5.1421E-2(5.1E-4)	5.0151E-2(3.1E-6)	5.0150E-2(3.4E-6)
DTLZ3	5.1829E-1(1.5E-1)	6.5375E-1(3.8E-1)	5.0856E-1(6.5E-1)	1.6290E+0(1.3E-0)	1.0267E+0(9.1E-1)	1.4211E+0(1.0E+0)
DTLZ4	4.6424E-1(3.4E-1)	2.8527E-1(3.2E-1)	1.7346E-1(2.3E-1)	5.1825E-2(1.1E-3)	1.1261E-1(1.7E-1)	5.0157E-2(1.3E-5)
DTLZ5	2.5627E-2(1.1E-3)	2.4306E-2(1.6E-3)	1.8107E-2(1.5E-5)	1.8101E-2(9.7E-6)	3.1920E-2(1.0E-4)	3.1786E-2(1.2E-4)
DTLZ6	1.0148E-1(2.0E-2)	8.2460E-2(2.4E-2)	2.4922E-2(3.7E-2)	3.3996E-2(3.2E-2)	6.4541E-1(5.7E-2)	7.7428E-1(5.1E-2)
DTLZ7	4.2838E-1(2.9E-1)	3.4369E-1(2.3E-1)	1.2946E-1(9.6E-4)	1.2907E-1(1.2E-3)	1.2840E-1(9.3E-4)	1.2773E-1(4.7E-4)
UF1	4.6600E-2(3.4E-3)	4.1616E-2(6.1E-3)	2.1201E-2(3.3E-2)	1.6545E-3(2.2E-4)	7.4430E-2(6.9E-2)	1.1961E-2(2.6E-2)
UF2	4.6187E-2(1.9E-3)	2.3027E-2(2.8E-3)	1.4578E-2(2.2E-2)	6.6129E-3(1.4E-2)	5.2665E-2(5.2E-2)	1.2393E-2(2.8E-3)
UF3	7.1289E-2(2.8E-2)	4.2602E-2(2.3E-2)	1.1098E-2(1.3E-2)	9.8210E-3(7.6E-3)	3.8405E-2(3.1E-2)	2.4841E-2(2.7E-2)
UF4	5.2290E-2(2.0E-3)	5.1971E-2(2.7E-3)	6.4407E-2(4.0E-3)	6.5795E-2(5.7E-3)	6.7517E-2(5.7E-3)	6.6706E-2(6.1E-3)
UF5	1.1661E+0(1.2E-1)	5.1095E-1(1.4E-1)	4.9510E-1(1.7E-1)	4.3255E-1(1.6E-1)	4.6553E-1(1.3E-1)	5.0649E-1(1.7E-1)
UF6	3.5670E-1(2.1E-2)	2.8411E-1(1.1E-1)	5.0766E-1(1.5E-1)	4.7432E-1(1.7E-1)	5.2782E-1(1.4E-1)	5.1456E-1(1.7E-1)
UF7	2.3288E-2(1.9E-3)	1.9814E-2(1.5E-3)	2.0330E-2(1.5E-2)	1.2150E-2(9.9E-3)	4.8084E-2(1.2E-1)	1.9857E-2(1.2E-2)
UF8	3.8577E-1(9.3E-3)	3.6610E-1(2.5E-2)	5.4927E-2(1.5E-2)	6.0981E-2(1.4E-2)	1.1147E-1(4.6E-2)	9.2471E-2(3.2E-2)
UF9	9.9491E-2(3.0E-3)	1.2012E-1(4.3E-2)	1.1732E-1(4.8E-2)	1.3119E-1(3.4E-2)	1.3193E-1(3.7E-2)	1.2819E-1(4.1E-2)
UF10	2.2497E+0(2.8E-1)	2.3864E+0(1.8E-1)	4.6931E-1(6.6E-2)	4.6540E-1(8.1E-2)	4.8214E-1(1.1E-1)	4.8285E-1(8.9E-2)
DTLZ2(4)	1.7281E-1(1.1E-3)	1.7250E-1(9.1E-4)	8.9093E-2(7.0E-4)	8.9121E-2(7.9E-4)	8.7399E-2(5.1E-6)	8.7399E-2(4.8E-6)
DTLZ2(6)	3.7149E-1(2.4E-3)	3.7347E-1(2.7E-3)	2.8757E-1(1.4E-2)	2.8751E-1(1.2E-2)	2.5665E-1(2.4E-5)	2.5665E-1(1.9E-5)
DTLZ2(10)	8.6901E-1(3.4E-2)	5.6541E-1(2.1E-3)	5.6890E-1(1.9E-2)	5.7893E-1(1.6E-2)	4.9222E-1(2.4E-5)	4.9222E-1(2.5E-5)
DTLZ5(2,10)	7.4584E-1(0.0E+0)	1.0511E-2(2.3E-3)	1.6977E-1(1.8E-3)	1.7116E-1(1.0E-3)	6.4940E-2(5.3E-5)	6.4930E-2(4.2E-5)
DTLZ5(3,10)	6.1236E-1(7.6E-2)	2.7412E-1(1.2E-1)	2.5113E-1(3.2E-4)	2.5110E-1(3.7E-4)	1.6867E-1(3.1E-3)	1.6739E-1(3.9E-3)

evolutions benefit from the information share and exchange under the BCE framework.

B. Individual Exploration in the PC Evolution

In BCE, the individual exploration operation plays a key role. It is designed to compensate for the possible diversity loss of the NPC evolution by exploring some promising individuals in the PC population that are not well developed or have already been eliminated in the NPC population. In this section, we take a closer look at this operation, investigating its role during the evolutionary process. That is, we record

how many individuals are produced in the operation and from them, how many individuals enter the PC and NPC populations via the PC and NPC selection, respectively.

For a comparison observation, we consider two situations in which the embedded non-Pareto algorithm performs poorly and well, separately. They are BCE-IBEA and BCE-MOEA/D+PBI on DTLZ2 [Fig. 6(a) and (c)]. Fig. 15 gives the evolutionary trajectories of the average number of those individuals that are produced in the exploration operation and enter the PC and NPC populations across the 30 runs. As can be seen in Fig. 15, the exploration appears to be adaptive, based on the performance of the embedded non-Pareto algorithm. If the embedded

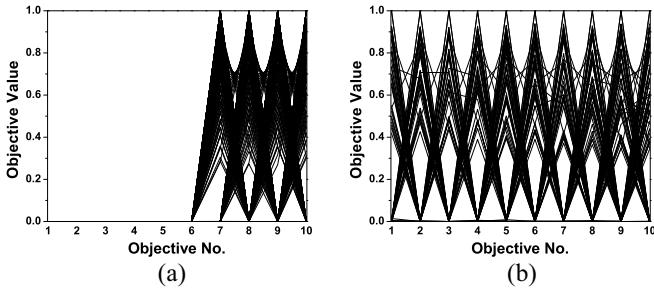


Fig. 14. Final solution set of IBEA and the one working under the BCE framework (i.e., the NPC population of BCE-IBEA) on the ten-objective DTLZ2, as shown by parallel coordinates. (a) IBEA. (b) BCE-IBEA.

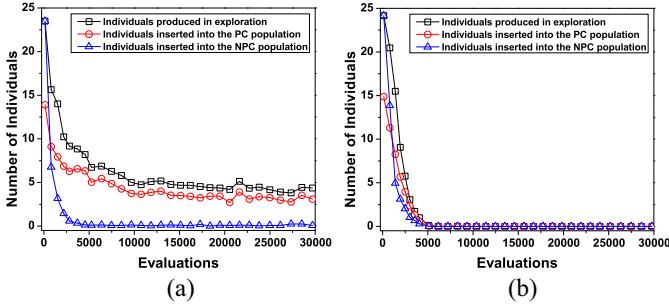


Fig. 15. Evolutionary trajectories of the average number of individuals that are produced in individual exploration and enter the PC and NPC populations across the 30 runs of (a) BCE-IBEA and (b) BCE-MOEA/D+PBI on DTLZ2. Black square denotes the number of individuals produced in the individual exploration operation, and red circle and blue triangles denote the number of the individuals entering the PC and NPC populations, respectively.

algorithm performs poorly, constant exploration is being made throughout the whole evolutionary process; if the algorithm works well, the exploration stops at certain evaluations, giving the NPC evolution more computational resources. This adaptive operation leads to a good balance between the NPC and the PC evolutions during the search process and enables BCE to be always competitive no matter whether the embedded non-Pareto algorithm performs well or not.

Next, we consider the number of individuals that enter the PC and NPC populations. In both situations, most of the individuals produced in the exploration operation are preserved in the PC population. This shows the effectiveness of the exploration in producing competitive individuals in terms of the PC. On the other hand, very few individuals produced in the exploration operation can be selected into the NPC population after around 3000 evaluations for BCE-IBEA because the NPC evolution of the algorithm already performs “well” based on its own criterion. In spite of that, there do exist a number of individuals successfully entering the NPC population in the initial stage of the evolution for both algorithms, especially at the first generation where all the individuals produced in the exploration operation are preserved in the NPC population. This indicates the effectiveness of exploring nondominated individuals on accelerating the evolution of the NPC population during the initial stage of the search.

C. Population Maintenance in the PC Evolution

Like in Pareto-based algorithms, the population in the PC evolution is maintained by the Pareto dominance relation and

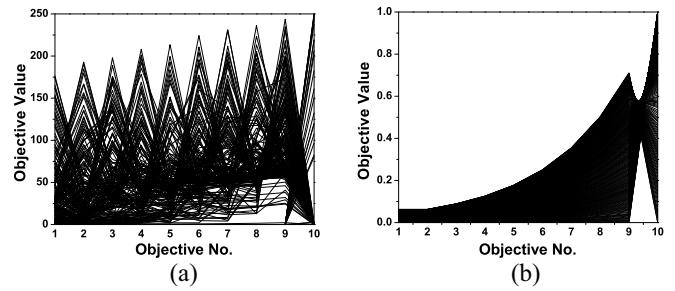


Fig. 16. Final solution set obtained by (a) NSGA-II and (b) BCE-MOEA/D+TCH on DTLZ5(2,10), as shown by parallel coordinates.

individual density. They prefer nondominated individuals and individuals with a lower crowding degree. Now a concern may arise: does the PC evolution suffer from what Pareto-based algorithms commonly suffer, such as inferior performance on MOPs with a complex PS [44] or with a high-dimensional objective space [68]?

In fact, the answer to the above question can be found from the results in the previous section (Sections IV-F and IV-G). As shown in Table III, for most of the variable-linkage and many-objective problems, the PC population outperforms the solution set obtained by the indicator-based and decomposition-based algorithms. And these non-Pareto algorithms have already been demonstrated to have a clear advantage over Pareto-based algorithms on such MOPs [25], [32], [44], [51], [68], [76]. This suggests a fundamental difference of performance between the PC evolution and the Pareto-based algorithms.

For a visual comparison, we give the results of BCE-MOEA/D+TCH and a well-known Pareto-based algorithm, NSGA-II, on the ten-objective problem DTLZ5(2,10). Fig. 16 plots the solution set obtained by the two algorithms via parallel coordinates. It is clear that in contrast to NSGA-II whose solutions are far away from the optimal front (the objective value being up to around 200), BCE-MOEA/D+TCH performs superiorly, with its solutions fully covering the whole Pareto front. This contrast indicates the interplay between the NPC evolution and the PC evolution in the BCE framework. Not only does the PC evolution maintain a set of well-distributed individuals to compensate for possible diversity loss of the NPC population but the NPC evolution also guides the PC population forward—it produces sufficient well-converged individuals, which can “pull” the PC population toward the Pareto front.

Finally, it is necessary to point out that although BCE generally works well on the MOPs where Pareto-based algorithms have struggled, its Pareto dominance and density-based population maintenance strategy, in some cases, still has an impact on the algorithm’s performance. This maintenance strategy can cause the existence of some dominance resistant solutions⁵ (DRSs) [30] in the PC population. This has often been observed in problems with many local optimal

⁵DRSs are the solutions with a quite poor value in at least one of the objectives but with (near) optimal values in the others, which Pareto-based algorithms have difficulty in getting rid of [18], [30], [36].

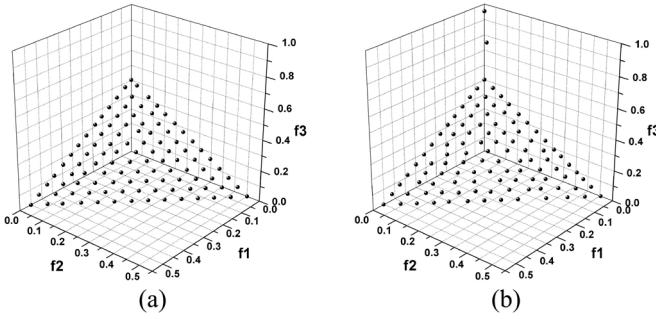


Fig. 17. Final solution set of (a) MOEA/D+TCH and (b) BCE-MOEA/D+TCH on DTLZ1.

fronts, such as DTLZ1 and DTLZ3. Fig. 17 shows the final solution set obtained by MOEA/D+TCH and BCE-MOEA/D+TCH in one typical run on DTLZ1. Clearly, in contrast to MOEA/D+TCH whose solutions all converge into the Pareto front, there exist two solutions far away from the optimal front in BCE-MOEA/D+TCH. Such solutions typically have a low crowding degree and will be preferred since no individual in the population dominates them.

The existence of DRSs in the PC population is detrimental not only to population maintenance but also to individual exploration. In the individual exploration operation, DRSs are always considered since there is no NPC individual located in their niche. Exploring them could have very little contribution to the algorithm's performance in view of their poor performance in terms of convergence. A straightforward approach to remove DRSs is to increase the selection pressure of Pareto dominance; however, this will lead to nondominated individuals to be treated differently, and thus will probably affect their distribution uniformity over the Pareto front. We leave this for our future study.

D. Comparison With State-of-the-Art Algorithms

The previous experimental results have demonstrated the effectiveness of the BCE framework in improving three non-Pareto algorithms. In this section, we further investigate the competitiveness of the BCE framework by comparing the three BCE algorithms with two state-of-the-art EMO algorithms, NSGA-III [16] and SMS-EMOA [7].

NSGA-III and SMS-EMOA use both PC and NPC in their selection mechanism. NSGA-III combines the Pareto nondominated sorting with a decomposition-based niching technique to balance solutions' convergence and diversity in the evolutionary process. NSGA-III has shown its advantage over the two decomposition-based algorithms MOEA/D-TCH and MOEA/D-PBI in its original paper [16] and has been found to significantly outperform IBEA in a recent study [69]. Working with the Pareto nondominated sorting, SMS-EMOA maximizes the HV contribution of nondominated solutions during the evolutionary process. SMS-EMOA has also been demonstrated to generally outperform IBEA and MOEA/D in a very recent study [39].

The intention that we introduce NSGA-III and SMS-EMOA as peer algorithms is to 1) verify the competitiveness of BCE in comparison with hybrid algorithms based on both PC and

NPC and 2) see how the three BCE algorithms would perform against state-of-the-art algorithms that outperform their original non-Pareto versions.

Note that the execution of SMS-EMOA with a large population size and a large number of objectives can take unacceptable time. Therefore, for some MOPs [i.e., UF8–UF10, many-objective DTLZ2 and DTLZ5(I, m)], we approximately estimate the HV indicator in SMS-EMOA by the Monte Carlo sampling method used in [4]. Following the practice in HypE [4], 10 000 sampling points are used. In addition, all configurations in this experiment were kept the same as in previous studies.

Table VI gives the experimental results of the three BCE algorithms against NSGA-III and SMS-EMOA. As can be seen, the three BCE algorithms generally outperform NSGA-III and SMS-EMOA. Specifically, BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI perform statistically better than, equal to, or worse than NSGA-III on 28/8/6, 27/6/9, and 21/5/16 problems, respectively. BCE-IBEA, BCE-MOEA/D+TCH, and BCE-MOEA/D+PBI perform statistically better than, equally to, or worse than SMS-EMOA on 21/6/15, 23/3/16, and 21/3/18 problems, respectively.

It is worth mentioning that actually the original versions of the three non-Pareto algorithms (i.e., IBEA, MOEA/D+TCH, and MOEA/D+PBI) are significantly outperformed by NSGA-III and SMS-EMOA. From the comparison of the IGD results, IBEA, MOEA/D+TCH, and MOEA/D+PBI perform statistically better than, equal to, or worse than NSGA-III on 11/4/27, 15/3/24, and 13/4/25 problems, respectively; IBEA, MOEA/D+TCH, and MOEA/D+PBI perform statistically better than, equally to, or worse than SMS-EMOA on 5/3/34, 13/4/25, and 11/8/23 problems, respectively. This contrast clearly indicates the effectiveness of the BCE framework—when working under the BCE framework, all three non-Pareto algorithms have a significant performance improvement and now are very competitive with or even generally outperform the state-of-the-art NSGA-III and SMS-EMOA.

VI. DISCUSSION

One important issue in the proposed BCE framework is the setting of the niche radius since both the population maintenance and individual exploration operations involve the niche-based density estimation. BCE considers the average of the Euclidean distance from all the individuals to their k th nearest individual in the population as the niche radius. A large k will result in a large radius. However, how to set k cannot be treated as trivial.

In population maintenance, the crowding degree estimation of an individual is affected by the number of other individuals in its niche (called its neighbors). A large k would make outer individuals of the population to be preferred since the number of their neighbors is generally fewer than that of inner ones. A too small k would make many individuals have no neighbor residing in their niche, thereby leading to the failure of differentiating them. In fact, the BCE algorithms can work well in terms of diversity maintenance when $k \in [3, 6]$. Fig. 18 plots the solution sets obtained by

TABLE VI
IGD RESULTS (MEAN AND SD) OF THE THREE BCE ALGORITHMS, NSGA-III, AND SMS-EMOA. THE TWO MARKS ASSOCIATED WITH EACH BCE ALGORITHM INDICATE ITS STATISTICAL COMPARISON (ACROSS 30 RUNS) AGAINST NSGA-III AND SMS-EMOA, RESPECTIVELY. “<,” “≈,” AND “>” INDICATE THAT THE BCE ALGORITHM STATISTICALLY PERFORMS BETTER, EQUALLY, AND WORSE, RESPECTIVELY, AT A 0.05 LEVEL BY THE WILCOXON’S RANK SUM TEST

Problem	BCE-IBEA	BCE-MOEA/D+TCH	BCE-MOEA/D+PBI	NSGA-III	SMS-EMOA
SCH1	1.6732E-2(1.1E-4) <<	1.6761E-2(1.4E-4) <<	1.6729E-2(9.3E-5) <<	4.8446E-2(2.0E-4)	1.8529E-2(3.2E-4)
SCH2	2.0902E-2(2.3E-4) <>	2.0832E-2(2.7E-4) <>	2.0861E-2(2.1E-4) <>	3.0851E-2(2.4E-3)	1.9640E-2(1.3E-4)
KUR	3.4693E-2(6.7E-4) <>	3.4142E-2(6.2E-4) <>	3.4960E-2(9.2E-4) <>	4.0147E-2(6.6E-4)	3.2104E-2(3.4E-4)
ZDT1	3.9475E-3(5.0E-5) <>	4.2756E-3(9.4E-5) >	5.9401E-3(4.3E-4) >	4.0427E-3(5.7E-5)	3.6445E-3(1.9E-5)
ZDT2	4.0420E-3(3.3E-4) ≈	4.1688E-3(9.3E-5) >	7.2119E-3(7.8E-4) >	4.0919E-3(1.2E-4)	4.3563E-3(1.8E-4)
ZDT3	4.6028E-3(6.5E-5) ≈	4.7997E-3(6.0E-5) ≈	5.4836E-3(1.5E-4) ≈	5.5924E-3(1.4E-4)	1.0180E-2(1.2E-2)
ZDT4	1.2270E-1(5.9E-2) >	8.2586E-3(3.2E-3) <<	1.3458E-2(6.5E-3) <<	4.7561E-2(4.0E-2)	2.3873E-2(3.9E-2)
ZDT6	3.4449E-3(8.6E-5) <<	1.0328E-2(1.3E-3) <>	2.1787E-2(2.7E-3) >	1.2670E-2(1.2E-3)	3.8964E-3(1.5E-4)
WFG1	7.7944E-1(6.3E-2) <<	9.6279E-1(5.4E-2) <<	< 1.0490E+0(5.3E-2) <<	1.1094E+0(6.4E-2)	1.1023E+0(1.3E-1)
WFG2	2.5660E-2(1.1E-2) ≈	2.0238E-2(4.1E-3) ≈	3.0526E-2(6.9E-3) >	2.5672E-2(3.7E-3)	1.8368E-2(3.9E-3)
WFG3	1.6083E-2(2.0E-3) <>	2.3228E-2(3.3E-3) >	3.9358E-2(7.4E-3) >	2.1876E-2(3.7E-3)	1.4330E-2(1.2E-3)
WFG4	1.3005E-2(4.6E-4) <<	1.9664E-2(1.5E-3) >	3.8183E-2(3.9E-3) >	1.6868E-2(1.1E-3)	1.1092E-2(2.7E-4)
WFG5	6.7165E-2(1.7E-4) <>	6.8558E-2(6.3E-4) ≈	7.7588E-2(2.1E-3) >	6.8390E-2(5.3E-4)	6.6540E-2(6.9E-5)
WFG6	6.1760E-2(9.2E-3) ≈	6.4056E-2(8.2E-3) ≈	8.6534E-2(1.2E-2) >	6.7004E-2(8.3E-3)	5.9407E-2(7.8E-3)
WFG7	1.4264E-2(3.1E-4) <>	1.5242E-2(3.3E-4) <>	2.9628E-2(2.2E-3) >	1.6195E-2(2.9E-4)	1.1408E-2(7.8E-5)
WFG8	8.0848E-2(6.1E-3) <>	8.5839E-2(5.6E-3) <>	1.0653E-1(8.1E-3) >	8.9378E-2(4.7E-3)	7.7623E-2(4.9E-3)
WFG9	7.4934E-2(5.2E-2) ≈	5.6954E-2(4.5E-2) ≈	7.3108E-2(3.9E-2) ≈	5.7340E-2(4.8E-2)	3.7320E-2(4.4E-2)
VNT1	1.2731E-1(3.3E-3) <>	1.2240E-1(2.7E-3) <>	1.2544E-1(2.7E-3) <>	1.9823E-1(4.0E-2)	1.0868E-1(2.2E-2)
VNT2	1.2219E-2(3.2E-4) <<	1.2121E-2(2.8E-4) <<	1.2436E-2(2.5E-4) <<	2.4536E-2(4.4E-3)	1.3633E-2(7.0E-5)
VNT3	3.8870E-2(1.4E-3) <<	3.8536E-2(1.2E-3) <<	3.8244E-2(1.1E-3) <<	2.4137E-1(1.9E-1)	2.3838E-1(3.9E-2)
DTLZ1	2.2154E-2(2.0E-3) ≈	2.1149E-2(1.8E-3) ≈	2.2134E-2(2.4E-2) ≈	2.2708E-2(7.1E-3)	1.9170E-2(8.2E-5)
DTLZ2	5.3051E-2(1.0E-3) <>	5.1708E-2(7.7E-4) <>	5.1369E-2(8.3E-4) <>	5.0930E-2(3.7E-4)	7.2422E-2(6.9E-4)
DTLZ3	5.3253E-1(4.3E-1) <<	1.6158E+0(1.3E+0) ≈	1.4062E+0(1.1E+0) ≈	2.1496E+0(1.3E+0)	2.6143E+0(8.9E-1)
DTLZ4	2.3409E-1(3.4E-1) >	5.2207E-2(9.4E-4) >	5.1944E-2(9.8E-4) >	5.0989E-2(4.4E-4)	5.5827E-1(4.8E-1)
DTLZ5	4.1949E-3(2.4E-4) <<	4.4611E-3(3.2E-4) <<	4.5173E-3(3.4E-4) <<	1.1340E-2(1.7E-3)	5.1063E-3(2.0E-4)
DTLZ6	7.3724E-2(2.5E-2) <<	3.9000E-1(3.3E-2) >	7.6720E-1(4.8E-2) >	5.2645E-1(4.1E-2)	9.1480E-2(9.1E-3)
DTLZ7	2.7161E-1(2.5E-1) ≈	5.6166E-2(2.1E-3) <<	5.8486E-2(9.7E-4) <<	1.1630E-1(1.4E-2)	7.3461E-2(8.2E-4)
UF1	3.5263E-2(5.4E-3) ≈	1.6430E-3(1.6E-4) <<	9.3327E-3(2.1E-2) <<	3.3575E-2(3.4E-3)	3.3638E-2(9.8E-4)
UF2	2.2549E-2(2.8E-3) <<	6.5645E-3(1.4E-3) <<	1.1159E-2(2.7E-3) <<	4.3056E-2(2.2E-3)	3.9715E-2(1.1E-3)
UF3	4.2563E-2(2.3E-2) <<	9.5736E-3(1.0E-2) <<	1.3594E-2(1.5E-2) <<	7.8088E-2(2.2E-2)	1.0859E-1(8.0E-3)
UF4	5.0860E-2(2.6E-3) >	6.6063E-2(5.7E-3) >	6.6512E-2(5.9E-3) >	4.9504E-2(2.2E-3)	5.0426E-2(1.1E-3)
UF5	5.1121E-1(1.4E-1) <<	4.0341E-1(1.4E-1) <<	4.7409E-1(1.5E-1) <<	1.1570E+0(1.2E-1)	1.1627E+0(3.9E-2)
UF6	2.8400E-1(1.1E-2) <<	4.2500E-1(1.4E-1) ≈	4.4913E-1(1.5E-1) ≈	3.9800E-1(1.9E-2)	3.8820E-1(9.9E-3)
UF7	1.5350E-2(1.1E-3) >	1.2116E-2(5.0E-3) <<	1.4598E-2(3.8E-3) <<	1.4883E-2(9.8E-4)	1.4730E-2(2.7E-4)
UF8	1.3722E-1(3.5E-2) ≈	5.6104E-2(2.1E-2) <<	8.9914E-2(3.3E-2) <<	1.2600E-1(3.5E-3)	1.2589E-1(5.1E-3)
UF9	1.1081E-1(4.1E-2) ≈	1.3153E-1(3.4E-2) >	1.2588E-1(4.1E-2) >	1.1108E-1(6.0E-3)	9.8454E-2(4.3E-3)
UF10	2.3864E+0(1.8E-1) <<	4.6496E-1(7.0E-2) <<	4.2631E-1(6.1E-2) <<	2.5358E+0(2.9E-1)	2.4508E+0(7.7E-2)
DTLZ2(4)	9.8155E-2(4.8E-3) >	9.3172E-2(3.2E-3) >	9.3679E-2(4.6E-3) >	8.7617E-2(2.0E-4)	1.1543E-1(7.6E-4)
DTLZ2(6)	2.3931E-1(2.7E-3) <<	2.3766E-1(1.9E-3) <<	2.4064E-1(7.6E-4) <<	2.5721E-1(1.6E-4)	2.9009E-1(1.4E-3)
DTLZ2(10)	4.8722E-1(4.8E-3) <<	4.6966E-1(7.4E-3) <<	4.7015E-1(2.7E-3) <<	4.9596E-1(8.0E-4)	5.4540E-1(1.6E-3)
DTLZ5(2,10)	2.1183E-3(3.2E-5) <<	2.1620E-3(7.2E-5) <<	2.1492E-3(6.8E-5) <<	1.2713E+0(6.3E-1)	4.8257E-1(2.9E-1)
DTLZ5(3,10)	3.7996E-2(3.1E-3) <<	3.7136E-2(1.1E-3) <<	3.8918E-2(2.1E-3) <<	5.6242E-2(4.4E-3)	4.9480E-2(2.3E-3)

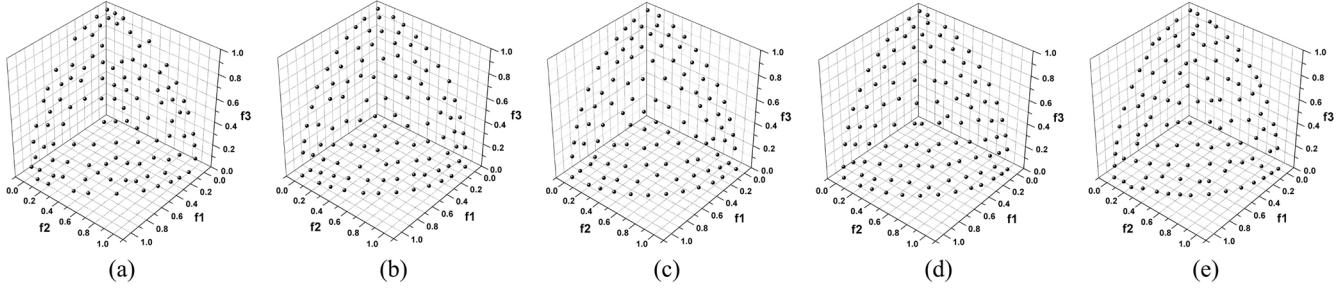


Fig. 18. Solution sets obtained by BCE-MOEA/D+TCH with different k values in the niche radius setting on DTLZ2. (a) $k = 2$. (b) $k = 3$. (c) $k = 4$. (d) $k = 6$. (e) $k = 10$.

BCE-MOEA/D+TCH with different k values on DTLZ2. As shown, BCE-MOEA/D+TCH with $k = 3, 4, 6$ performs well, whereas the algorithm with $k = 2$ struggles to maintain uniformity, and more boundary solutions are obtained when k is set to 10.

In individual exploration, the niche size affects the number of individuals to be explored. A large niche can lead to very few (or even none of) individuals in the PC population to be explored. Table VII gives the experimental results

of BCE-MOEA/D+TCH with different k values on the nine WFG problems. Similar results can also be observed on other problems. As can be seen from the table, setting a small k can generally lead to a better result of the algorithm. BCE-MOEA/D+TCH with $k = 2$ performs best or second best in seven out of the nine problems, and the algorithm with $k = 3$ in eight out of the nine problems. This indicates that setting k to 2 or 3 is suitable for the individual exploration operation.

TABLE VII
IGD RESULTS (MEAN AND SD) OF BCE-MOEAD+TCH WITH DIFFERENT k VALUES ON THE WFG PROBLEMS. THE BEST AND SECOND BEST MEANS FOR EACH PROBLEM ARE SHOWN WITH DARK AND LIGHT GRAY BACKGROUNDS, RESPECTIVELY

Problem	$k = 2$	$k = 3$	$k = 4$	$k = 6$
WFG1	9.3814E-1(6.1E-2)	9.6279E-1(5.4E-2)	9.8176E-1(6.0E-2)	9.8774E-1(5.5E-2)
WFG2	2.1396E-2(6.5E-3)	2.0238E-2(4.1E-3)	2.0754E-2(2.5E-3)	2.3346E-2(5.6E-3)
WFG3	2.1886E-2(3.0E-3)	2.3228E-2(3.3E-3)	2.4681E-2(6.9E-3)	2.3794E-2(3.6E-3)
WFG4	1.8452E-2(1.4E-3)	1.9664E-2(1.5E-3)	2.0035E-2(1.8E-3)	2.0269E-2(1.7E-3)
WFG5	6.8565E-2(5.1E-4)	6.8558E-2(6.3E-4)	6.8577E-2(5.1E-4)	6.8871E-2(8.2E-4)
WFG6	6.4045E-2(9.1E-3)	6.4056E-2(8.2E-3)	6.4176E-2(9.7E-3)	6.3763E-2(7.0E-3)
WFG7	1.5289E-2(4.8E-4)	1.5242E-2(3.3E-4)	1.5779E-2(4.0E-4)	1.5528E-2(4.1E-4)
WFG8	8.5569E-2(4.9E-3)	8.5839E-2(5.6E-3)	8.7812E-2(6.8E-3)	8.6347E-2(4.6E-3)
WFG9	6.0298E-2(4.7E-2)	5.6954E-2(4.5E-2)	6.6903E-2(4.8E-2)	5.8558E-2(4.4E-2)

From the above observations, the BCE algorithm with k set to 3 can work well in both the population maintenance and individual exploration operations.

Finally, it is worth mentioning that the previous experiments are all about the test of comprehensive performance of BCE (i.e., combined performance of convergence and diversity). Then, how does the BCE algorithm perform in terms of separate convergence or diversity? In fact, BCE is designed to use a non-Pareto algorithm (as a driver) to lead the PC evolution forward and at the same time use the PC evolution to compensate for the possible diversity loss during the search process of this non-Pareto algorithm. Thus, the BCE algorithm usually performs worse than the embedded non-Pareto algorithm in terms of convergence but better in terms of diversity. However, if the NPC used in the embedded algorithm struggles to steer the evolution forward, Pareto dominance would drive the evolution instead. In this case, the BCE algorithm has better convergence than the embedded non-Pareto algorithm. Fig. 7 in Section IV is precisely such a case.

VII. CONCLUSION

This paper has presented a BCE framework of PC and NPC to deal with MOPs. In BCE, the two criteria work collaboratively, attempting to use their strengths to facilitate each other's evolution. In general, the NPC evolution drives the PC evolution forward, whereas the PC evolution compensates for the possible diversity loss of the NPC evolution. In the proposed framework, the two populations communicate constantly, with their information being fully shared and compared in a generational manner. Any new individual produced in one population will be tested and applied in the other. The information comparison of the two populations reflects the current status of the NPC evolution, thus making the search more focused on some undeveloped (or not well-developed) but promising regions.

Systematic experiments have been carried out by investigating three representative non-Pareto EMO algorithms on seven categories of 42 test problems. The results have revealed the effectiveness of the BCE approach in providing a good balance between convergence and diversity. The three BCE algorithms work well, whether on problems in which NPC could struggle, such as MOPs with a highly irregular or a discontinuous Pareto front, or on problems in which the PC is likely to fail, such as MOPs with a complex PS or a high-dimensional objective space.

Moreover, the performance verification of the embedded non-Pareto algorithms indicates that both the PC and the NPC evolutions benefit from the information share and exchange under the BCE framework. In addition, two key operations of BCE, individual exploration and population maintenance, have been investigated and analyzed. The variation of the number of explored individuals during the evolutionary process has shown the adaptiveness of individual exploration, depending on the performance of the embedded algorithm. As to population maintenance, despite clear differences having been observed from the results in comparison with Pareto-based algorithms, the Pareto dominance and density-based maintenance strategy could have an impact on the performance of the BCE algorithm. Finally, a comparison with NSGA-III and SMS-EMOA has verified the competitiveness of the three BCE algorithms as independent algorithms to deal with MOPs.

The BCE of Pareto and non-Pareto criteria is a general framework in EMO. It can be especially of practical value in the area, given its applicability for any non-Pareto algorithm, no requirement of parameter tuning in the implementation, and the reliability on various problems with distinct characteristics.

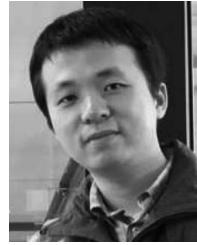
Finally, note that the study in this paper focuses on the design of the BCE framework and the implementation of selection operations, whereas the variation operation is not fixed and it uses the same search operators from the embedded non-Pareto algorithm. In the subsequent work, we will attempt to introduce other search operators into BCE. This includes integrating existing operators (such as those from MO-CMA-ES [29], MTS [65], MTS2 [11], and SBS [46]) or designing new operators specially for the individual exploration in the PC evolution.

REFERENCES

- [1] N. Al Moubayed, A. Petrovski, and J. McCall, “ D^2MOPSO : MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces,” *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, 2014.
- [2] M. Asafuddoula, T. Ray, and R. Sarker, “A decomposition-based evolutionary algorithm for many objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
- [3] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, “Theory of the hypervolume indicator: Optimal μ -distributions and the choice of the reference point,” in *Proc. 10th ACM SIGEVO Workshop Found. Genet. Algorithms*, Orlando, FL, USA, 2009, pp. 87–102.
- [4] 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.

- [5] M. Basseur and E. K. Burke, "Indicator-based multi-objective local search," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 3100–3107.
- [6] R. P. Beausoleil, "MOSS: Multiobjective scatter search applied to nonlinear multiple criteria optimization," *Eur. J. Oper. Res.*, vol. 169, no. 2, pp. 426–449, 2006.
- [7] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [8] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner, "Approximation-guided evolutionary multi-objective optimization," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 1198–1203.
- [9] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the R2 indicator," in *Proc. Genet. Evol. Comput. Conf.*, Philadelphia, PA, USA, 2012, pp. 465–472.
- [10] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.
- [11] C. Chen and L.-Y. Tseng, "An improved version of the multiple trajectory search for real value multi-objective optimization problems," *Eng. Optim.*, vol. 46, no. 10, pp. 1430–1445, 2014.
- [12] C. A. C. Coello, "Evolutionary multiobjective optimization," *Wiley Interdiscipl. Rev. Data Mining Knowl. Disc.*, vol. 1, no. 5, pp. 444–447, 2011.
- [13] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. 9th Annu. Conf. Genet. Evol. Comput. Conf.*, London, U.K., 2007, pp. 773–780.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [15] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
- [16] 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.
- [17] 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.
- [18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Berlin, Germany: Springer, 2005, pp. 105–145.
- [19] A. Díaz-Manríquez, G. Toscano-Pulido, C. A. C. Coello, and R. Landa-Becerra, "A ranking method based on the R2 indicator for many-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 1523–1530.
- [20] T. Friedrich, C. Horoba, and F. Neumann, "Multiplicative approximations and the hypervolume indicator," in *Proc. 11th Annu. Conf. Genet. Evol. Comput. Conf.*, Montreal, QC, Canada, 2009, pp. 571–578.
- [21] S. B. Gee, X. Qiu, and K. C. Tan, "A novel diversity maintenance scheme for evolutionary multi-objective optimization," in *Proc. 14th Int. Conf. Intell. Data Eng. Autom. Learn.*, Hefei, China, 2013, pp. 270–277.
- [22] S. B. Gee, K. C. Tan, V. A. Shim, and N. R. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 542–559, Aug. 2015.
- [23] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 428–442.
- [24] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 5, pp. 3677–3688, 2012.
- [25] Z. He and G. G. Yen, "Ranking many-objective evolutionary algorithms using performance metrics ensemble," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2480–2487.
- [26] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.
- [27] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. C. Coello, and J. Molina, "Pareto-adaptive ϵ -dominance," *Evol. Comput.*, vol. 15, no. 4, pp. 493–517, 2007.
- [28] 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.
- [29] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.
- [30] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Seoul, Korea, 2001, pp. 957–962.
- [31] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.
- [32] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.
- [33] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms," in *Proc. 9th Int. Conf. Parallel Probl. Solving Nat.*, Reykjavik, Iceland, 2006, pp. 493–502.
- [34] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *Proc. 11th Int. Conf. Parallel Probl. Solving Nat.*, Kraków, Poland, 2010, pp. 91–100.
- [35] A. L. Jaimes, L. V. S. Quintero, and C. A. C. Coello, "Ranking methods in many-objective evolutionary algorithms," in *Nature-Inspired Algorithms for Optimisation*, R. Chiong, Ed. Berlin, Germany: Springer, 2009, pp. 413–434.
- [36] A. L. Jaimes, "Techniques to deal with many-objective optimization problems using evolutionary algorithms," Ph.D. dissertation, Dept. Comput. Sci., Center Res. Adv. Studies Nat. Polytech. Inst., Mexico City, Mexico, May 2011.
- [37] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. 7th Int. Conf. Nat. Comput.*, vol. 3. Shanghai, China, 2011, pp. 1260–1264.
- [38] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.
- [39] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.
- [40] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *Proc. 9th Int. Conf. Parallel Probl. Solving Nat.*, Reykjavik, Iceland, 2006, pp. 553–562.
- [41] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. Edinburgh, U.K., 2005, pp. 443–450.
- [42] 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.
- [43] H. Li and D. Landa-Silva, "An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 481–494.
- [44] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [45] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2014.
- [46] M. Li, S. Yang, K. Li, and X. Liu, "Evolutionary algorithms with segment-based search for multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1295–1313, Aug. 2014.
- [47] 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.
- [48] M. Li, S. Yang, and X. Liu, "A test problem for visual investigation of high-dimensional multi-objective search," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 2140–2147.
- [49] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45–65, Nov. 2015.
- [50] M. Li, S. Yang, and X. Liu, "A performance comparison indicator for Pareto front approximations in many-objective optimization," in *Proc. Genet. Evol. Comput. Conf.*, Madrid, Spain, 2015, pp. 703–710.
- [51] M. Li, S. Yang, X. Liu, and R. Shen, "A comparative study on evolutionary algorithms for many-objective optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 261–275.

- [52] M. Li, S. Yang, J. Zheng, and X. Liu, "ETEA: A Euclidean minimum spanning tree-based evolutionary algorithm for multiobjective optimization," *Evol. Comput.*, vol. 22, no. 2, pp. 189–230, 2014.
- [53] K. Lwin, R. Qu, and J. Zheng, "Multi-objective scatter search with external archive for portfolio optimization," in *Proc. 7th Int. Joint Conf. Comput. Intell.*, Vilamoura, Portugal, 2013, pp. 111–119.
- [54] S. Mostaghim and H. Schmeck, "Distance based ranking in many-objective particle swarm optimization," in *Proc. 10th Int. Conf. Parallel Probl. Solving Nat.*, Dortmund, Germany, 2008, pp. 753–762.
- [55] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optim.*, Zürich, Switzerland, 2001, pp. 82–95.
- [56] A. J. Nebro *et al.*, "AbYSS: Adapting scatter search to multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 439–457, Aug. 2008.
- [57] 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.
- [58] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [59] G. Rudolph, H. Trautmann, S. Sengupta, and O. Schütze, "Evenly spaced Pareto front approximations for tricriteria problems based on triangulation," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.*, Sheffield, U.K., 2013, pp. 443–458.
- [60] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim.*, Matsushima, Japan, 2007, pp. 5–20.
- [61] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.
- [62] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari, "Framework for many-objective test problems with both simple and complicated Pareto-set shapes," in *Proc. 6th Int. Conf. Evol. Multi-Criterion Optim.*, Ouro Preto, Brazil, 2011, pp. 197–211.
- [63] K. Sindhy, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.
- [64] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.
- [65] L.-Y. Tseng and C. Chen, "Multiple trajectory search for unconstrained/constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 1951–1958.
- [66] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Dept. Electr. Comput. Eng., Graduate School Eng., Air Force Inst. Technol., Dayton, OH, USA, 1999.
- [67] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proc. Annu. Conf. Genet. Evol. Comput. Conf.*, Amsterdam, The Netherlands, 2013, pp. 687–694.
- [68] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim.*, Matsushima, Japan, 2007, pp. 742–756.
- [69] H. Wang, L. Jiao, and X. Yao, "TwoArch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.
- [70] U. K. Wickramasinghe and X. Li, "Using a distance metric to guide PSO algorithms for many-objective optimization," in *Proc. 11th Annu. Conf. Genet. Evol. Comput. Conf.*, Montreal, QC, Canada, 2009, pp. 667–674.
- [71] 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.
- [72] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, to be published.
- [73] A.-C. Zăvoianu, E. Lughofer, G. Bramerdorfer, W. Amrhein, and E. P. Klement, "DECMO2: A robust hybrid and adaptive multi-objective evolutionary algorithm," *Soft Comput.*, vol. 19, no. 2, pp. 3551–3569, 2014.
- [74] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-487*, 2009.
- [75] 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.
- [76] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 203–208.
- [77] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
- [78] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design Optim. Control*, Barcelona, Spain, 2002, pp. 95–100.
- [79] 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.
- [80] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000.
- [81] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of Pareto set approximations," in *Multiobjective Optimization*, vol. 5252, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Berlin, Germany: Springer, 2008, pp. 373–404.
- [82] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Probl. Solving Nat.*, Birmingham, U.K., 2004, pp. 832–842.



Miqing Li received the B.Sc. degree in computer science from the School of Computer and Communication, Hunan University, Changsha, China, in 2004, and the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University, Xiangtan, China, in 2008. He is currently working toward the Ph.D. degree with the Department of Computer Science, Brunel University London, London, U.K.

He has published over 20 research papers. His research interests include evolutionary multiobjective and many-objective optimization algorithms and their application.



Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China, in 1993, 1996, and 1999, respectively.

He is a Professor of Computational Intelligence and the Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 200 publications. His research interests include evolutionary and genetic algorithms, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Dr. Yang is the Chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, under the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society, and the Founding Chair of the Task Force on Intelligent Network Systems, under the Intelligent Systems Applications Technical Committee of the IEEE Computational Intelligence Society.



Xiaohui Liu received the B.Eng. degree in computing from Hohai University, Nanjing, China, and the Ph.D. degree in computer science from Heriot-Watt University, Edinburgh, U.K.

He is a Professor of Computing with Brunel University London, London, U.K., where he directs the Centre for Intelligent Data Analysis. He has over 100 high-quality journal publications on computational intelligence and data science.

Prof. Liu was a recipient of the Highly Cited Researchers Award by Thomson Reuters.