A Fuzzy Decomposition based Multi/Many-objective Evolutionary Algorithm

Songbai Liu, Qiuzhen Lin, *Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, Maoguo Gong, *Senior Member, IEEE*, and Carlos A. Coello Coello, *Fellow, IEEE*

***Abstract*—Performance of multi/many-objective evolutionary algorithms (MOEAs) based on decomposition is highly impacted by the Pareto front (PF) shapes of multi/many-objective optimization problems (MOPs), as their adopted weight vectors may not properly fit the PF shapes. To avoid this mismatch, some MOEAs treat solutions as weight vectors to guide the evolutionary search, which can adapt to the target MOP's PF automatically. However, their performance is still affected by the similarity metric used to select weight vectors. To address this issue, this paper proposes a fuzzy decomposition based MOEA. At first, a fuzzy prediction is designed to estimate the population's shape, which helps to exactly reflect the similarities of solutions. Then, *N* least similar solutions are extracted as weight vectors to get *N* constrained fuzzy subproblems (*N* is the population size), and accordingly a shared weight vector is calculated for all subproblems to provide a stable search direction. At last, the corner solution for each of *m* least similar subproblems (*m* is the objective number) is preserved to maintain diversity, while one solution having the best aggregated value on the shared weight vector is selected for each of the remaining subproblems to speed up convergence. When compared to several competitive MOEAs in solving a variety of test MOPs, the proposed algorithm shows some advantages at fitting their different PF shapes.**

***Index Terms*—Evolutionary algorithm, Multi/many-objective optimization, Fuzzy decomposition.**

I. Introduction

EVOLUTIONARY algorithms characterized by a population-based iterative search engine have been recognized as an effective approach for solving multi/many-objective optimization problems (MOPs), which are found in some real-life applications [1]-[3]. In this paper, the unconstrained MOPs are considered, as modeled by

$$\text{Minimize } F(x) = (f_1(x),..., f_m(x)), \quad (1)$$
$$\text{Subject to: } x \in \Omega,$$

where $x = (x_1,\ldots, x_n)$ is an n dimensional vector in the decision space Ω and $F(x)$ includes m (often conflicting) objectives to be optimized. The term multi-objective optimization problem is used when m = 2 or 3 and many-objective optimization problem is used when $m > 3$. When considering all the objectives, a set of equally optimal solutions called Pareto set is often found, and the mapping of this set into the objective function space is called Pareto front (PF) [8]. The main goal of solving MOPs is to find a set of solutions that can closely and evenly approximate their PFs [4]-[7].

In recent years, numerous multi/many-objective evolutionary algorithms (MOEAs) have been proposed to solve various MOPs, which can be classified into three main categories, i.e., Pareto-based MOEAs [9]-[10], indicator-based MOEAs [11]-[13], and decomposition-based MOEAs (MOEADs) [14]-[15]. MOEADs have become very popular in recent years, mainly due to their promising performance and implementation efficiency [16]-[18]. With the decomposition approach, the target MOP is transformed into a set of subproblems, which are optimized in a collaborative way using evolutionary search. However, the performance of MOEADs will be highly affected by their adopted weight vectors or aggregation methods. Some relevant works on this topic are briefly reviewed below.

For the weight vectors used in MOEADs, they are first initialized to have an even distribution in objective space, which helps to maintain the population's diversity. In some early MOEADs [14]-[16], their weight vectors are uniformly sampled from the unit hyperplane $f_1 + f_2 + ... + f_m = 1$ by using a systematic extraction method, such as Das and Dennis's method [19], Deb and Jain's method [20] or mixture uniform design [21]. However, the performance of MOEADs strongly depends on the PF shapes, when the weight vectors with even distribution are used [22]. These MOEADs are good at solving MOPs with regular PFs, but perform poorly on MOPs whose PF shapes are degenerate, disconnected, inverted, or strongly convex/concave [28]. This is mainly due to the low matching degree of the evenly distributed weight vectors and the irregular PF shapes. To avoid this mismatching, three main kinds of improved approaches are used in some recent MOEADs, such as adaptive extraction of weight vectors [23]-[25], dynamic adjustment of weight vectors [26]-[32], and the use of multiple sets of weight vectors [33]-[34], adapting the distribution of weight vectors to fit various PF shapes.

Regarding the aggregation methods, choosing an effective aggregation function to formulate scalar subproblems can help

Manuscript received xx. xx. 2019; revised xx. xx. 2019; accepted xx. xx. 2019. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61876110 and 61836005, the Joint Funds of the NNFC under Key Program Grant U1713212, and CONACyT Grant no. 2016-01-1920. This work was also supported in part by the NSFC under Grant 61876162, and by the Shenzhen Scientific Research and Development Funding Program under Grants JCYJ20180307123637294 and JCYJ20190808164211203. (*Corresponding author: Qiuzhen Lin*)

S.B. Liu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Q.Z. Lin is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: qiuzhlin@szu.edu.cn).

K. C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen, China (e-mail: kaytan@cityu.edu.hk).

M.G. Gong is with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China.

C.A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evol. Comput. Group), México, D.F. 07300, MÉXICO.

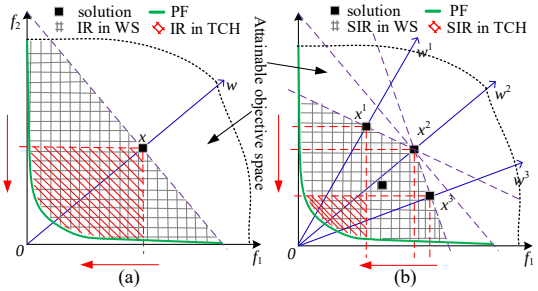


Fig. 1 Simple examples of (a) IR in WS and TCH, (b) SIR in WS and TCH

to balance convergence and diversity during the evolutionary search. Primitively, three aggregation functions, i.e., weighted sum (WS), Tchebycheff (TCH), and penalty-based boundary intersection (PBI), are introduced in [14], which are also used in some recent MOEA/Ds [35]-[36]. However, as discussed in [37]-[38], the search capabilities of these MOEADs are also highly impacted by their aggregation functions. For example, as shown in Fig. 1, the improvement region (IR) and the shared IR (SIR) for a subproblem decomposed from WS are always larger than that from TCH in the attainable objective space. Note that the detailed definitions of IR and SIR can be found in [40]-[41], where a larger size of IR implies a larger probability to find a solution associated to the subproblem with a better aggregated value, while a larger size of SIR indicates a larger probability to have the same better solutions for the subproblem and its neighbors (here, the neighbors of a subproblem are defined based on the distances between their relevant weight vectors). Thus, as observed in Fig. 1, the search capabilities of MOEADs using WS are generally better to speed up convergence and worse to maintain diversity when compared to that using TCH [40]-[41], especially on MOPs with a large number of objectives [39]. In order to better balance convergence and diversity with aggregation functions in MOEADs, three main kinds of improved methods are proposed, such as embedding constraints into aggregation functions [42]-[46], adaptive selection of aggregation functions [37], [47], and new aggregation functions [48].

Although the above efforts have been conducted on weight vectors or aggregation methods to further enhance MOEADs, it is not always an easy task to select a suitable aggregation method adhering with a specific set of weight vectors, which is expected to conform the target MOP's PF and characteristic for getting superior performance [41]. Recently, without using the pre-set weight vectors, solutions in some MOEAs [49]-[60] are treated as weight vectors, trying to guide the population to approximate the PF adaptively. Two metrics are often used to select these solutions as weight vectors, which can reflect their direction similarity. One is the direction angle of two solutions in the objective space, which is used in VaEA [49], MOEA/D-AM2M [51], MaOEA-CSS [50], PAEA [52], hpaEA [53], MaOEA/C [54], SPSAT [55], and PaRP/EA [56]. The other is the distance of two solutions' projections on the unit hyperplane $f_1 + f_2 + \ldots + f_m = 1$ (m is the objective number), which is adopted in DDEA [57] and MaOEA-DDFC [58]. A smaller value of the direction angle or the projections' distance on the unit hyperplane means a higher direction similarity for two solutions, which is also used to define the neighborhoods of

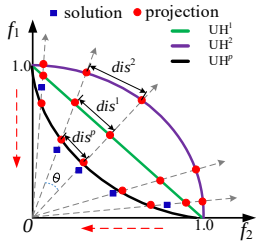


Fig. 2 Direction similarity between solutions measured with the distance between their projections on UH^1, UH^2 and an appropriate UH^p

subproblems [14]. In Fig. 2, a simple example is plotted to show the measurement of direction similarity based on the direction angle (denoted by θ) or the projections' distance (denoted by $dis^{0.5}$, dis^1 and dis^2) on three different unit hyperplanes UH^p with p=0.5, 1 and 2 (here UH^p indicates the unit hyperplane $f_1^p + f_2^p + \ldots + f_m^p = 1$, where p is a positive parameter to determine the curvature of the hyperplane). Then, in these MOEAs [49]-[56], solutions with high direction similarity and poor convergence are removed, while those with low direction similarity are added into the population to compensate their diversity. Thus, the search directions are automatically guided by solutions, which have superior performance on solving various MOPs [49]-[58]. It is worth noting that, the direction angle θ in Fig. 2 is actually equal to another special metric dis^2, as they will return the same comparison results on the direction similarity. Thus, the most commonly used metrics θ and dis^1 in these MOEAs [49]-[56] are actually two special cases in the family metrics dis^p that represent the projections' distance on UH^p with p=1 and p=2. Obviously, as illustrated in Fig. 2, it is insufficient to use the above two metrics to measure the direction similarity of solutions [54], but an adaptive metric with variable dis^p should be adopted when tackling MOPs with different PF shapes.

Given the above discussions, a fuzzy decomposition based multi/many-objective evolutionary algorithm (called FDEA) is proposed in this paper for solving MOPs with various PF shapes. In FDEA, a fuzzy decomposition is proposed to divide the target MOP as N constrained fuzzy subproblems (N is the population size), which includes two main components, i.e., a fuzzy prediction and a weight vector extraction. Here, fuzzy prediction is used to roughly predict a suitable p for UH^p that fuzzily fits the current non-dominated solution set. Then, the direction similarity between solutions can be appropriately measured based on their projections on UH^p. After that, the weight vector extraction is used to select N solutions as weight vectors based on the direction similarity, which can keep a high matching degree with the target MOP's PF. Furthermore, one shared weight vector can be obtained from the N extracted weight vectors, which provides a stable search direction for all fuzzy subproblems. At last, an elite selection strategy is run, which chooses one corner solution for each of the m least similar subproblems (m is the number of objectives) to maintain diversity and then selects one solution with the best aggregated fitness value for each of other subproblems to ensure convergence. When compared to some competitive MOEAs, our algorithm has shown some advantages in solving numerous test MOPs with various PF shapes.

The remainder of this paper is organized as follows. Section

II introduces our motivations to design FDEA. Section III presents the details of FDEA. The experimental results and some discussions are provided in Section IV. At last, our conclusions and future work are presented in Section V.

II. Motivations

In early studies of MOEADs [35]-[36], their weight vectors are evenly sampled from the unit hyperplane, resulting in the fact that their performance is sensitive to the matching degree of weight vectors and the PF shapes [22]. To alleviate this problem, solutions are selected as weight vectors based on the direction similarity in some MOEAs [49] [50] [51] [54] [56] [52], trying to automatically guide the evolutionary search toward different PFs. However, there still exist some open challenges for these MOEAs.

First, there are only two special metrics for calculating direction similarity in these MOEAs, i.e., dis^1 and θ (equal to dis^2) in Fig. 2, which actually represent the projections' distance on UH^p with p=1 and p=2. These two metrics cannot eliminate the performance sensitivity when solving different MOPs with complicated PF shapes. For example, the angle θ between two solutions (equal to dis^2) is applied in VaEA [49], while dis^1 is implemented in DDEA [56]. When using VaEA and DDEA to solve the 3-objective MaF3 problem with a convex PF [59], their final solutions in a single run will be distributed mainly on the central area of the PF, while the PF boundaries are rarely covered, as illustrated in Fig. 3. This is reasonable since, as pointed out in MaOEA/C [54], the metric θ or dis^2 is not very appropriate to reflect the distribution of solutions near the boundaries of the convex PF, which can be observed in Fig. 4(a). Six solutions x^1 to x^6 actually have a similar performance on convergence and diversity, but θ_1 (the angle between x^1 and x^2) and θ_5 (the angle between x^5 and x^6) are much smaller than θ_2 (the angle between x^3 and x^4), which indicates that x^3 and x^4 show better distribution based on the metric θ when compared to other solutions near the boundaries. Similarly, if dis^1 is used for tackling convex MOPs, all solutions are projected onto the UH^1 to calculate their distances. However, this kind of unfair phenomenon still exists for the boundary solutions (see Fig. 4(a)). Thus, an appropriate metric to exactly reflect the direction similarity between solutions is very important for improving the performance of these MOEAs. In PAEA [52] and PaRP/EA [56], the direction similarity is modified to compute the angles of two solutions on two adversarial directions, i.e., one emanating from the ideal point and the other backward from the nadir point (z^{nad}), as plotted in Fig. 4(b). However, the angle oriented from z^{nad} for two solutions is still a special case of their projections' distance on a unit hypersurface.

Second, solutions are selected as weight vectors in some MOEAs [49]-[54], leading to the fact that a dynamic change of weight vectors happens in each generation. As pointed out in [26]-[31], the frequent change of weight vectors may deteriorate the convergence speed, as the dynamically changed search directions in each generation may be confusing. To alleviate this problem, the weight vectors in MOEA/D-AM2M [51] are only changed once in each of τ generations (τ is set to 50 in its experiments), while a fixed m-D weight vector (1, 1,…, 1) (m is the objective number) is used with the WS method in VaEA [49], MaOEA/C [54], PAEA [52] and DAEA [57], which can provide a more stable search direction in the evolutionary process.

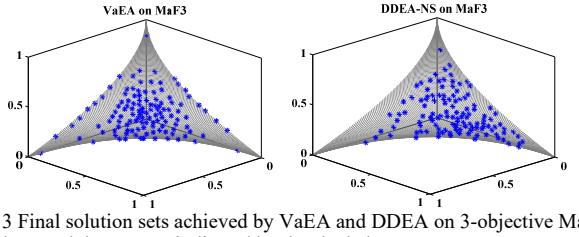


Fig. 3 Final solution sets achieved by VaEA and DDEA on 3-objective MaF3 problem, and the true PF indicated by the shaded area.

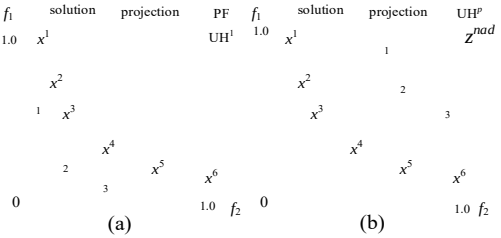


Fig. 4 Illustration of (a) the dis^1 and dis^2 (or the angle θ) between boundary solutions and intermediate solutions for the convex PF, and (b) the nadir point oriented angle between solutions.

Based on the above discussions, in order to extract weight vectors that can be well adapted to various PF shapes, a natural idea is to fuzzily predict the target MOP's PF shape during the evolutionary process, which can help to obtain an appropriate metric to exactly reflect the direction similarity. Then, weight vectors should be extracted to fit the PF shapes and an elite selection strategy should be designed to balance convergence and diversity during the evolutionary process, under the case that the weight vectors are dynamically changed in each generation. Thus, to cover all the above mentioned issues, FDEA is proposed in this paper with fuzzy decomposition and elite selection, which will be introduced in the following section.

III. The Proposed Algorithm

The proposed FDEA algorithm includes two main components, i.e., fuzzy decomposition and elite selection. At first, after the offspring population is generated, fuzzy decomposition is executed, which includes a fuzzy prediction to estimate the population's shape and a weight vector extraction to define N constrained fuzzy subproblems (N is the population size). After that, one shared weight vector is further obtained using the N extracted weight vectors, which can provide a stable search direction for all subproblems. At last, an elite selection strategy is run to get the next population, which selects one corner solution for each of m least similar subproblems (m is the number of objectives) and one solution with the best aggregated value for each of the rest subproblems. By this way, our algorithm can properly balance convergence and diversity for solving MOPs with different PF shapes. To introduce FDEA, its main framework is first introduced in Section III.A, and then the details of fuzzy decomposition and elite selection are respectively given in Section III.B and III.C to clarify the implementation of FDEA.

A. The Main Framework

Here, to give an overview of FDEA, its main framework is provided in **Algorithm 1** with three inputs: N (the population size), m (the number of objectives), and $G_{\max}$ (the maximum number of generations). In line 1, an initial population $\boldsymbol{P}$ is randomly generated to have N solutions in decision space Ω for solving the target MOP, and the generation counter G is initialized as 1. Then, an offspring population $\boldsymbol{Q}$ is produced to have N new solutions in line 3, by running variation operations (i.e., simulated binary crossover (SBX) [60] and polynomial-based mutation (PM) [61]) on N parent pairs randomly selected from $\boldsymbol{P}$. Then, these two populations ($\boldsymbol{P}$ and $\boldsymbol{Q}$) are merged together in line 4 to get a union population $\boldsymbol{U}$ including $2N$ solutions. Next, an adaptive normalization procedure used in NSGA-III [20] is run on $\boldsymbol{U}$ to mitigate the impact of different scaled objectives in MOPs, as shown in line 5. Thus, the ith objective $f_i(x)$ of each solution x in $\boldsymbol{U}$ is normalized as

$$f_i'(x)=\frac{f_i(x)-z_i^*}{z_i^{nad}-z_i^*}, \tag{2}$$

where $f_i'(x)$ ($i=1,2,...,m$) indicate the ith normalized objective of x, while $z^*=(z_1^*,z_2^*,...,z_m^*)$ is an ideal point and $z^{nad}=(z_1^{nad},z_2^{nad},...,z_m^{nad})$ is a nadir point, which are obtained according to all the non-dominated solutions in $\boldsymbol{U}$ using the estimation method in [20], [56]. After that, in line 6, our fuzzy decomposition method is run to decompose the target MOP into N constrained fuzzy subproblems, which will be introduced in Section III.B. Then, the union population $\boldsymbol{U}$ will be divided to get N subsets ($\boldsymbol{S_1}$, $\boldsymbol{S_2}$,…, $\boldsymbol{S_N}$) respectively associated to N constrained fuzzy subproblems, by gathering solutions with high direction similarity in each subset. At last, the elite selection process is run in line 7 to select one solution from each of subsets $\boldsymbol{S_1}$, $\boldsymbol{S_2}$,…, $\boldsymbol{S_N}$ to compose the new population, which will be introduced in Section III.C. Then, G is increased by 1 in line 8. While G is smaller than $G_{\max}$, the above procedures in lines 3-8 will be run iteratively; otherwise, $\boldsymbol{P}$ is reported as the final approximate solutions for the target MOP in line 10.

To further clarify the running of FDEA, the details of fuzzy decomposition and elite selection are introduced in the following subsections.

B. Fuzzy Decomposition

In our fuzzy decomposition, a fuzzy prediction is first run to estimate the PF shape and then a weight vector extraction is executed to get N weight vectors, which define N constrained fuzzy subproblems. To clarify the running of fuzzy decomposition, its pseudo-code is given in **Algorithm 2** with three inputs: $\boldsymbol{U}$ (the union population), N (the population size), and m (the objective number). In line 1, $\boldsymbol{S_A}$ (collecting all non-dominated solutions), $\boldsymbol{S_B}$ (collecting all candidate solutions), and N subsets $\boldsymbol{S_i}$ (collecting all solutions associated to i-th subproblem, i=1, 2,…,N) are all initialized as empty sets. After running the non-dominated sorting [9] on $\boldsymbol{U}$ in line 2, $\boldsymbol{U}$ is divided into L solution subsets ($\boldsymbol{F_1}$, $\boldsymbol{F_2}$,…, $\boldsymbol{F_L}$) based on their non-dominated ranks in $\boldsymbol{U}$. Then, set $\boldsymbol{S_A}$ as $\boldsymbol{F_1}$, and $\boldsymbol{S_B}$ is constructed in lines 3-5 by including one subset each time, staring from $\boldsymbol{F_1}$, then $\boldsymbol{F_2}$ and so on, until the size of $\boldsymbol{S_B}$ exceeds N for the first time. Let $\boldsymbol{F_l}$ be the last subset included into $\boldsymbol{S_B}$, i.e., $\boldsymbol{S_B}$ $=\boldsymbol{F_1}+\boldsymbol{F_2}+\ldots+\boldsymbol{F_l}$. The remaining fronts of $\boldsymbol{U}$ (i.e., $\boldsymbol{F_{l+1}}$ to $\boldsymbol{F_L}$) are not considered in this process. After that, our proposed fuzzy prediction is run on $\boldsymbol{S_A}$ in line 6, aiming to find an appropriate unit hypersurface (UH^p) that fuzzily fits the shape of the current non-dominated solutions. The details of fuzzy prediction will be introduced in Section III.B.1. After the estimation of PF shape, each solution $x\in\boldsymbol{S_B}$ will get a projection x' on UH^p in line 7, which is an intersection of a direction ray (from the origin pointing to x) on UH^p. Then, the ith normalized objective of x' can be obtained as follows:

$$f_i'(x')=\frac{f_i'(x)}{\left[\sum_{k=1}^{m}[f_k'(x)]^p\right]^{1/p}}, \tag{3}$$

where $f_i'(x)$ ($i=1,2,...,m$) is the ith normalized objective of x in (2). Thus, the direction similarity of two solutions in $\boldsymbol{U}$ can be defined as the distance of their projections on UH^p. Then, a weight vector extraction introduced in Section III.B.2 is run in line 8 to select N solutions from $\boldsymbol{S_B}$ as weight vectors (saved in $\boldsymbol{S_W}$), while the remaining solutions in $\boldsymbol{S_B}$ are preserved in $\boldsymbol{S_R}$. Consequently, N constrained fuzzy subproblems are defined in lines 9-11, by treating the ith solution $x^i\in\boldsymbol{S_W}$ as the first candidate solution for ith subproblem and adding x^i into $\boldsymbol{S_i}$ (i = 1, 2,…,N). Afterwards, each solution in $\boldsymbol{S_R}$ will be associated to the subproblem with the highest direction similarity. Specifically, the distances of jth solution y^j (j = 1, 2,…,$|\boldsymbol{S_R}|$) in $\boldsymbol{S_R}$ to ith weight vector x^i (i = 1, 2,…,N) in $\boldsymbol{S_W}$ are calculated in line 14, the weight vector u with the highest direction similarity for y^j is found in line 16, and then y^j is added into the subset $\boldsymbol{S_u}$ to associate with its subproblem in line 17. At last, N solution subsets ($\boldsymbol{S}_1,\boldsymbol{S}_2,...,\boldsymbol{S}_N$) respectively associated to N sub-

Algorithm 1 FDEA(N, m, $G_{\max}$)

```
1: initialize P and set G=1
2: while G ≤ G_max do
3:    generate the offspring population Q
4:    U = P ⋃ Q and set P = ∅ , Q = ∅
5:    normalize all solutions in U
6:    (S_1, S_2,…, S_N) = Fuzzy-Decomposition(U, N, m)
7:    P = Elite-Selection(S_1,S_2,…,S_N)
8:    G++
9: end while
10: return P
```

Algorithm 2 *Fuzzy-Decomposition*($\boldsymbol{U}$, N, m)

```
1: initialize S_i = ∅ , i = 1,2,...,N , S_A = ∅ , S_B = ∅
2: divide U into multiple subsets F_1,F_2,…,F_L by the fast
   non-dominated sorting, and set S_A = F_1, l = 1
3: while |S_B| < N
4:    S_B = S_B ⋃ F_l, l++
5: end while
6: p = Fuzzy-Prediction(S_A)
7: solutions in U are mapped on UH^p to get the projections
8: (S_W, S_R) = Weight-Vector-Extraction(S_B, N, m)
9: for i = 1 to N
10:   add the ith solution in S_W into S_i
11: end for
12: for j = 1 to |S_R|
13:   for i = 1 to N
14:      compute DS(y^j, x^i) with (11), y^j ∈ S_R and x^i ∈ S_W
15:   end for
16:      associate u = k : arg min_{k=1 to N} DS(y^j, x^k)
17:      add y^j into S_u
18: end for
19: return (S_1, S_2,..., S_N)
```

Algorithm 3 *Fuzzy-Prediction*($\boldsymbol{S_A}$)

1: remove solutions outside the unit hypercube from $\boldsymbol{S_A}$
2: initialize $E = 0, \sigma = 0, p = 1$
3: **for** each $x \in \boldsymbol{S_A}$
4: compute $Dis^1(x)$ with (4)
5: **end for**
6: compute E with (5)
7: compute σ with (6)
8: preset two samples sets sp_1 and sp_2 of p_s with (7) //step 1
9: calculate $Fit(p_s)$ by (8) for each sample p_s //step 2
10: preliminarily predict p by (9) //step 3
11: fuzzily adjust p by (10) //step 4
12: **return** p

problems are returned in line 19.

To further clarify the running of this fuzzy decomposition, the details of fuzzy prediction and weight vector extraction are respectively introduced below.

1) Fuzzy Prediction

In this process, an appropriate coefficient p on UH^p is estimated to effectively guide the evolutionary search and our fuzzy prediction is limited in the unit hypercube as defined by the origin and the m-dimensional point (1, 1,…,1). The pseudo-code of fuzzy prediction on UH^p is given in **Algorithm 3** with the input: $\boldsymbol{S_A}$ with all non-dominated solutions. In line 1, solutions outside the unit hypercube are first removed from $\boldsymbol{S_A}$. Then, in line 2, the coefficient p is initialized as 1, while the average value (E) and standard deviation (σ) of all distances from the solutions in $\boldsymbol{S_A}$ to the linear unit hypersurface UH^1 are initialized as 0. After that, in lines 3-6, the distance of each solution $x \in \boldsymbol{S_A}$ to UH^1 (marked as $D^1(x)$) can be computed as

$$D^1(x) = \frac{\sum_{i=1}^{m} f_i'(x) - 1}{\sqrt{m}}, \tag{4}$$

where $f_i'(x)$ is the ith normalized objective of x in (2) and m is the objective number. Please note that the distance $Dis^1(x)$ in (4) can be positive or negative. Specifically, the solution x below UH^1 will have $Dis^1(x) < 0$, otherwise will get $Dis^1(x) \geq 0$. Different cases of $Dis^1(x)$ are shown in Fig. 5, where $Dis^1(x^1) < 0$ and $Dis^1(x^2) > 0$. Then, in lines 6 and 7, the values of E and σ can be respectively calculated by (5) and (6), as follows:

$$E = \frac{\sum_{x \in S_A} Dis^1(x)}{|\boldsymbol{S_A}|}, \tag{5}$$

$$\sigma = \sqrt{\frac{\sum_{x \in S_A} (Dis^1(x) - E)^2}{|\boldsymbol{S_A}| - 1}}, \tag{6}$$

where $|\boldsymbol{S_A}|$ denotes the cardinality of $\boldsymbol{S_A}$. Based on E and σ, p can be roughly predicted in lines 8-11 by the following steps:

Step1: preset two sample sets (sp_1 and sp_2) of p', as follows:

$$p' = \begin{cases} 1 - 0.05i, & p' \in sp_1 \text{ and } i = 0,1,...,T_1 - 1 \\ 1 + 0.1j, & p' \in sp_2 \text{ and } j = 0,1,...,T_2 - 1 \end{cases}, \tag{7}$$

where p' is a sample value of p, while sp_1 and sp_2 contain T_1 and T_2 samples of p', respectively. Here, T_1=17 and T_2=51.

Step 2: calculate the average fitting value of $\boldsymbol{S_A}$ and $UH^{p'}$ for each p' from sp_1 and sp_2, denoted as $Fit(p')$, as follows:

$$Fit(p') = \frac{\sum_{i=1}^{|\boldsymbol{S_A}|} \left[\sum_{j=1}^{m} [f_j'(x^i)]^{p'} \right]^{1/p'}}{|\boldsymbol{S_A}|}. \tag{8}$$

Step 3: preliminarily predict p as a suitable p' with the closest

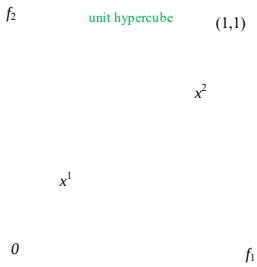


Fig. 5 Distance between solutions and UH^1 in the unit hypercube, where $Dis^1(x^1) < 0$.

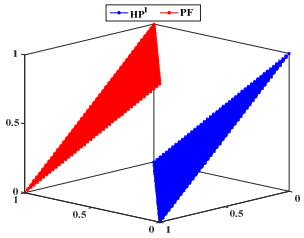


Fig. 6 Example of a MOP's PF with a linear shape but far away from UH^1.

fitting between $UH^{p'}$ and $\boldsymbol{S_A}$, as follows:

$$p = \begin{cases} p' \in sp_1 : \arg\min \left| Fit(p') - 1 \right| & \text{if } E < 0, \\ p' \in sp_2 : \arg\min \left| Fit(p') - 1 \right| & \text{otherwise} \end{cases}. \tag{9}$$

Thus, if E>0, then $p \geq 1$, UH^p tends to be concave; E<0 or $E \to 0$ respectively indicate UH^p tends to be convex or linear.

Step 4: fuzzily adjust the value of p based on the coefficient of variation (cv) between E and σ, as follows:

$$p = \begin{cases} 1.0 + cv & \text{if } |cv| < 0.1 \wedge r_1 < 0.9 \\ p + r_3 & \text{if } |cv| > 0.1 \wedge r_2 > 0.9 \\ p & \text{otherwise} \end{cases}, \tag{10}$$

where $cv = \sigma / E$, r_1 and r_2 are two random numbers in [0,1], and r_3 is a small random disturbance in [-0.02,0.02]. The fuzzy adjustment of p here is mainly used to fit the MOPs with linear PF shapes that may be far away from the UH^1, as shown in Fig. 6. Thus, if cv approaches to 0, the value of p is predicted close to 1 with a high probability.

As discussed in Section II, in some angle-based MOEAs [49]-[56] and decomposition-based MOEAs [35]-[36], the PFs of MOPs are only approximated by UH^2 or UH^1 to define the distance of solutions. Moreover, the PF shapes of MOPs are estimated as the model of UH^p in $pa\lambda$-MyDE [73] and RIB-EMOA [74], where a new optimization problem is defined to find the most suitable value of p. However, the estimation methods need the true PF information to obtain the hypervolume (HV) [63], which are unpractical and inefficient on computational cost. Furthermore, other complicated models are studied by using some machine learning methods to exactly estimate the PF shapes of MOPs, e.g., the generic front model in [75] and growing neural gas network in [68]. However, their performance is significantly affected by the qualities of data for training and testing. Especially, in the early and median evolutionary stages, the solutions' qualities are relatively poor to match the PF shape, which may mislead the used machine learning methods. In a recently proposed MDEA [76], the PFs of MOPs are estimated by predicting a p value from a sample set {0.1, 0.2, …, 4} to calculate the Minkowski

Algorithm 4 *Weight-Vector-Extraction*($\boldsymbol{S_B}$, N, m)

1: initialize $\boldsymbol{S_W} = \varnothing$ and $\boldsymbol{S_R} = \varnothing$
2: **while** $|\boldsymbol{S_B}| > N$
3: get two most similar individuals $(x^d, x^q) \in \boldsymbol{S_B}$ with (13)
4: compute $DS(x^d, \boldsymbol{S_B})$ and $DS(x^q, \boldsymbol{S_B})$ with (12)
5: **if** $DS(x^d, \boldsymbol{S_B}) < DS(x^q, \boldsymbol{S_B})$
6: add x^d into $\boldsymbol{S_R}$ and remove x^d form $\boldsymbol{S_B}$
7: **else**
8: add x^q into $\boldsymbol{S_R}$ and remove x^q form $\boldsymbol{S_B}$
9: **end if**
10: **end while**
11: find two least similar individuals $(x^l, x^s) \in \boldsymbol{S_B}$ with (14)
12: add x^l and x^s into $\boldsymbol{S_W}$, and remove them from $\boldsymbol{S_B}$
13: **while** $|\boldsymbol{S_W}| < m$
14: get $x = x^t \in \boldsymbol{S_B} : \arg\max DS(x^t, \boldsymbol{S_W})$
15: add x into $\boldsymbol{S_W}$, and remove x from $\boldsymbol{S_B}$
16: **end while**
17: $\boldsymbol{S_W} = \boldsymbol{S_W} \bigcup \boldsymbol{S_B}$
18: **return** ($\boldsymbol{S_W}$, $\boldsymbol{S_R}$)

distance. Here, m extreme solutions and m+1 center solutions are identified at first, and then the fitting degrees between each UH^p sample and each pair of (center solution, extreme solution) are calculated in order to find the p value for the most frequently matched PF shape. When compared to the mentioned methods above, our approach is easily implementable and efficient in terms of computational cost, as it only needs to fuzzily fit the shape of current non-dominated solutions to the model of UH^p within some sample values of p.

2) Weight Vector Extraction

After the fuzzy prediction of UH^p is given in **Algorithm 3**, a number of solutions are selected from $\boldsymbol{S_B}$ based on the direction similarity of their projections on UH^p, which are then used as weight vectors to assist the fuzzy decomposition for the target MOP. Here, the pseudo-code of weight vector extraction is introduced in **Algorithm 4** with the inputs: $\boldsymbol{S_B}$ (all candidate solutions), N (the population size), and m (the objective number). Here, two definitions of calculating the direction similarity are given below.

Definition 1: Given two different solutions $x, y \in \boldsymbol{S_B}$, the direction similarity of x and y (termed $DS(x, y)$) is measured by the distance of their projections (i.e., x' and y') on UH^p, which can be computed as

$$DS(x,y) = \sqrt{\sum_{i=1}^{m} \left(f_i'(x') - f_i'(y') \right)^2}, \tag{11}$$

where $f_i'(x')$ and $f_i'(y')$ can be obtained by (3). A smaller value of $DS(x, y)$ means a higher direction similarity of x and y.

Definition 2: Given a solution $x \in \boldsymbol{S_B}$ and a subset $\boldsymbol{S_b} \subseteq \boldsymbol{S_B}$, the direction similarity between x and $\boldsymbol{S_b}$ (termed $DS(x, \boldsymbol{S_b})$) is measured by finding a solution $y \in \boldsymbol{S_b}$ with the minimum value of $DS(x, y)$ in $\boldsymbol{S_b}$. Thus, $DS(x, \boldsymbol{S_b})$ is set the same with $DS(x, y)$, which can be computed as follows:

$$DS(x, \boldsymbol{S_b}) = \underset{y \in \boldsymbol{S_b}, x \neq y}{\text{minimize}}\ DS(x,y), \tag{12}$$

A smaller value of $DS(x, \boldsymbol{S_b})$ means a higher direction similarity of x and $\boldsymbol{S_b}$. Note that (12) is also used in the recently proposed PMEA [77], which adaptively replaces each invalid weight vector by finding the solution vector with the lowest direction similarity to the remaining unselected solutions.

In line 1 of **Algorithm 4**, $\boldsymbol{S_W}$ (collecting the selected solutions as weight vectors) and $\boldsymbol{S_R}$ (collecting the remaining solutions) are all initialized as empty sets. Then, solutions as well as weight vectors are collected into $\boldsymbol{S_R}$ in lines 2-10, while the remaining solutions are collected into $\boldsymbol{S_W}$ in lines 11-17. In lines 2-10, the solution with the highest direction similarity to $\boldsymbol{S_B}$ is removed from $\boldsymbol{S_B}$ and added into $\boldsymbol{S_R}$ in lines 2-10, until the size of $\boldsymbol{S_B}$ is equal to N. To be specific, two solutions $(x^d, x^q) \in \boldsymbol{S_B}$ with the highest direction similarity are found in line 3, as follows:

$$(x^d, x^q) = \underset{(x^i, x^j) \in \boldsymbol{S_B}, i \neq j}{\arg\min}\ DS(x^i, x^j), \tag{13}$$

where $(d, q) \in \{1, 2, ..., |\boldsymbol{S_B}|\}$ and $DS(x^i, x^j)$ is computed by (11). Then, their direction similarity to $\boldsymbol{S_B}$ is computed by (12) in line 4. One solution with the higher direction similarity to $\boldsymbol{S_B}$ is removed from $\boldsymbol{S_B}$ and added into $\boldsymbol{S_R}$ in lines 5-8. After repeating the above process, $\boldsymbol{S_R}$ is obtained and $\boldsymbol{S_B}$ has N selected solutions.

In lines 11-16, m corner solutions [72] (i.e., the first m members of $\boldsymbol{S_W}$) are selected from $\boldsymbol{S_B}$ to maintain the boundary information of population. At first, two solutions $(x^l, x^s) \in \boldsymbol{S_B}$ with the lowest direction similarity in $\boldsymbol{S_B}$ are found in line 11 as the first two corner solutions, as follows:

$$(x^l, x^s) = \underset{(x^i, x^j) \in \boldsymbol{S_B}, i \neq j}{\arg\max}\ DS(x^i, x^j), \tag{14}$$

where $(i, j) \in \{1, 2, ..., |\boldsymbol{S_B}|\}$ and $DS(x^i, x^j)$ is computed by (11). Then, x^l and x^s are added into $\boldsymbol{S_W}$ and removed from $\boldsymbol{S_B}$ in line 12. By this way, $\boldsymbol{S_W}$ has two selected solutions. While the number of solutions in $\boldsymbol{S_W}$ is smaller than m (i.e., $|\boldsymbol{S_W}| < m$) in line 13, one solution x with the lowest direction similarity to $\boldsymbol{S_W}$ will be found in line 14, which will be added into $\boldsymbol{S_W}$ and removed from $\boldsymbol{S_B}$ in line 15. Obviously, the iterative process of lines 14-15 will ultimately save m corner solutions in $\boldsymbol{S_W}$, and the remaining solutions in $\boldsymbol{S_B}$ are also added into $\boldsymbol{S_W}$ in line 17 in an ascending order. At last, the final solution sets $\boldsymbol{S_W}$ and $\boldsymbol{S_R}$ are returned in line 18 to assist the fuzzy decomposition described in **Algorithm 2**.

Algorithm 5 *Elite-Selection* ($\boldsymbol{S_1}$, $\boldsymbol{S_2}$, …, $\boldsymbol{S_N}$)

1: initialize $\boldsymbol{P} = \varnothing$, $w = (w_1, w_2, ..., w_m) = (1, 1, ..., 1)$
2: **for** i = 1 to m do
3: update w_i with (15)
4: **end for**
5: **for** i = 1 to m do
6: add $x^1 \in \boldsymbol{S_i}$ into $\boldsymbol{P}$, x^1 is the first individual of $\boldsymbol{S_i}$
7: **end for**
8: **for** $j = m$+1 to N do
9: get the optimal solution x for the jth subproblem by (16)
10: add x into $\boldsymbol{P}$
11: **end for**
12: **return** $\boldsymbol{P}$

C. Elite Selection

After the above fuzzy decomposition, N subsets are obtained as $\boldsymbol{S_1}, \boldsymbol{S_2}, ..., \boldsymbol{S_N}$, where solutions in each $\boldsymbol{S_i}$ are associated to ith subproblem (i=1,2,…,N). Based on the output of **Algorithm 2**, the first member of $\boldsymbol{S_i}$ is the weight vector for the ith subproblem, the first member in the first m subsets is also a corner solution for its subproblem, and each solution in $\boldsymbol{S_i}$ has a high direction similarity to the first member of $\boldsymbol{S_i}$.

In this process, an elite selection method is run to get N final solutions with balanceable convergence and diversity from

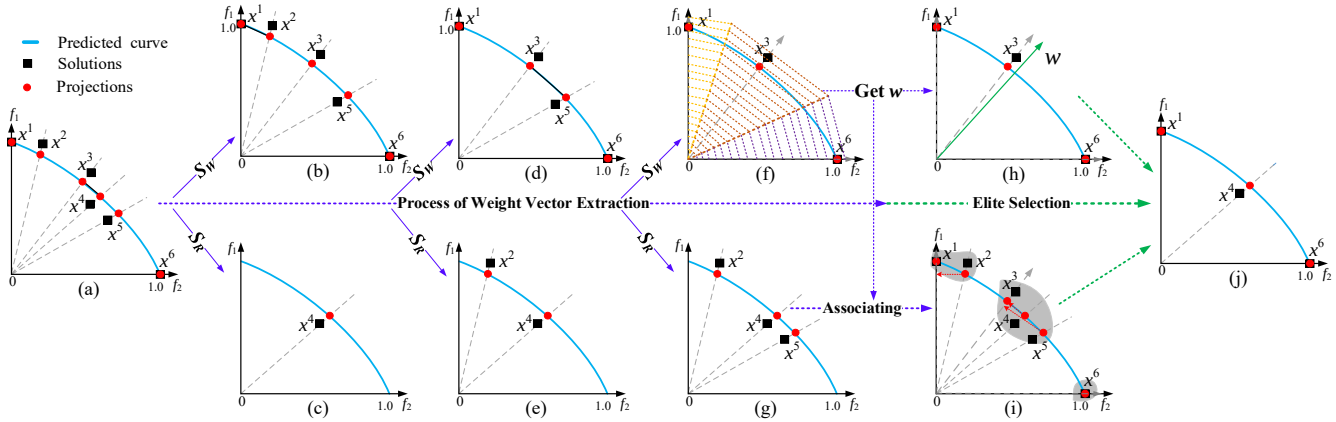


Fig.7 A simple example to show the process of FDEA, (a) for the fuzzy prediction, (b)-(g) for the weight vector extraction, and (h)-(j) for the elite selection.

$\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_N$, where one solution is selected from each subset. As shown in **Section III.B**, the population's diversity has been well maintained by N selected weight vectors, i.e., the first solution in each $\boldsymbol{S}_i$ (i=1,2,…,N). Thus, this elite selection prefers to select one solution with good convergence from each $\boldsymbol{S}_i$. In general, MOEADs will use an aggregated function (like WS, TCH or PBI) and the associated weight vectors to evaluate the solutions' quality, aiming to balance convergence and diversity when a number of evenly distributed weight vectors are used during the evolutionary process. However, N weight vectors extracted from **Algorithm 4** in FDEA are dynamically changed in each generation, which will highly affect the search direction if the traditional decompostion methods in [14]-[15] are used.

Here, in order to provide a stable search direction for each subproblem, a shared weight vector w and the WS aggregated function are used for all subproblems. The pseudo-code of elite selection is given in **Algorithm 5** with the inputs: N subsets ($\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_N$). In line 1, the new population $\boldsymbol{P}$ is initialized as an empty set and the shared weight vector w = $(w_1, w_2, ..., w_m)$ is initialized as $(1,1,...,1)$. In lines 2-4, each member w_k (k=1,2,…, m) of w is obtained by

$$w_k = \frac{\sum_{i=1 \text{ to } N, x^1 \in S_i} f_k'(x^{1'})}{N} \qquad (15)$$

where x^1 represents the first member of $\boldsymbol{S}_i$, $x^{1'}$ is the projection of x^1 on the predicted UH^p, and $f_i'(x^{1'})$ can be computed by (3). Please note that the fixed weight vector $(1,1,...,1)$ was used in some MOEAs [49]-[54] to guide the evolutionary search. However, an adaptive weight vector w applied here is more suitable, as it can follow the distribution of N extracted weight vectors, which can be more effective to guide the evolutionary search, especially for some imbalance cases in MOPs [27]. Thus, the jth constrained fuzzy subproblem can be mathematically defined as follows:

$$\text{Minimize } g^{ws}(x|r^j) = w_1 f_1'(x) + ... + w_m f_m'(x), \qquad (16)$$
$$\text{subject to: } x \in S_j,$$

where the weight vector r^j is the first solution in S_j, $f_k'(x)$ ($k = 1,2,...,m$) is the kth normalized objective of x in (2), the shared weight vector w is obtained by (15), and $j = 1,2,...,N$. With the assistance of the fuzzily predicted UH^p, the weight vector r^j of the jth subproblem is obtained by **Algorithm 4**, which is stored respectively as the jth member of $\boldsymbol{S}_W$ and the first member of $\boldsymbol{S}_j$. Specifically, the weight vectors $r^1, r^2, ..., r^N$ are adopted to define the constrained attainable objective subspace of each subproblem, get the associted solution set $\boldsymbol{S}_j$ in lines 9-18 of **Algorithm 2**, and obtain the shared w in (15). Next, m corner solutions are added into $\boldsymbol{P}$ by collecting the first member of $\boldsymbol{S}_i$ (i = 1, 2,…, m) in lines 5-7, which maintains the boundary information of population. Then, the currently best solution for each subproblem can be selected from its corresponding candidated solution set $\boldsymbol{S}_j$ (j =m+1, m+2,…, N) using (16). At last, the new population $\boldsymbol{P}$ having N currently optimial solutions for these N constrained fuzzy subproblems is returned in line 12 for the next generation.

D. Discussions

In the above subsections, the general framework and main components of FDEA have been introduced in detail. Here, a simple example is plotted in Fig. 7, which shows the process of FDEA on solving a minimization problem with two objectives (f_1 and f_2). In Fig. 7(a), assume that the union population U has 3 parents and 3 children (x^1 to x^6) that are non-dominated with each other. By using the fuzzy prediction in Algorithm 2, the predicted curve UH^p can be obtained in Fig. 7(a), and each solution correspondingly gets a projection on UH^p. Then, the direction similarity of any two different solutions is measured based on their projections' distance using (3). Three weight vectors for decomposition in (16) are extracted from U by iteratively dividing U into two subpopulations S_W and S_R, which are shown in Fig. 7(b) to Fig. 7(g). Concretely, the most similar solution pair (x^3, x^4) is identified in Fig. 7(a) using (13), followed by adding x^4 to S_R as $DS(x^3, U)$ is larger than $DS(x^4, U)$, as shown in Fig. 7(b) and Fig. 7(c). By the same way, the most similar solution pairs (x^1, x^2) in Fig. 7(b) and (x^3, x^5) in Fig. 7(d) are respectively found. Then, x^2 and x^5 are respectively added into S_R in Fig. 7(e) and Fig. 7(g). Thereafter, two least similar solutions x^1 and x^6 in Fig. 7(f) are treated as two corner solutions using (14), which are also the first two solutions of S_W. Thus, three solutions are saved in S_W, i.e., x^1, x^6 and x^3 in Fig. 7(f), which are treated as weight vectors to define three constrained attainable objective subspaces in Fig. 7(f) and to get a shared weight vector w in Fig. 7(h). Furthermore, three constrained fuzzy subproblems formulated by (16) are associated to three candidate solution sets, respectively, i.e.,

$\{x^1, x^2\}$, $\{x^6\}$, and $\{x^3, x^4, x^5\}$ in Fig. 7(i). Finally, one best solution for each subproblem is selected from its candidate solution set by the elite selection in Algorithm 5. As shown in Fig. 7(j), the selected three optimal solutions are x^1, x^6, and x^4.

Obviously, the proposed FDEA is a decomposition-based MOEA, where N weight vectors are extracted from the combined population to formulate N constrained fuzzy subproblems (N is the population size). Compared with the traditional decomposition-based MOEAs, FDEA is characterized with the following three features.

(1) FDEA involves a fuzzy prediction procedure before the decomposition of a MOP, which aims to estimate the shape of the current non-dominated solutions using the model UH^p, as plotted in Fig.7(a). Consequently, FDEA can handle MOPs with different curvatures in their PFs, e.g., convexity, linearity, or concavity.

(2) FDEA does not require a set of predefined weight vectors, but fuzzily extracts them from the population, which aims to decompose a MOP automatically, as plotted in Figs.7 (b)-7(g). Thus, FDEA can handle MOPs with irregular shapes of PFs, e.g., degenerated, inverted, and disconnected PFs.

(3) FDEA decomposes a MOP into N constrained fuzzy subproblems by (16), where each subproblem has a candidate solution set in its constrained attainable objective subspace (as plotted in Fig. 7(i)) and all subproblems share the same aggregation function (i.e., the WS function with the shared w in (15)) in order to provide a stable evolutionary direction, as plotted in Fig. 7(h). Thus, by collaboratively optimizing the shared WS function and getting the optimal solution for each subproblem from its candidate solution set, FDEA can balance the population's diversity and convergence well.

IV. Experimental Studies

A. Benchmark Problems and Performance Metrics

To investigate the effectiveness of our proposed FDEA, especially on problems with irregular PF shapes, a total of 30 test problems with different PF shapes are selected from the WFG [62], WFG4x [27] and MaF [59] test suites, including WFG1-WFG9, WFG41-WFG48, and MaF1-MaF13. In this study, the objective number m is set from 2 to 15, i.e., $m \in \{2,3,5,8,10,15\}$. The number of decision variables n in each problem is set as follows. For WFG1-WFG9 and WFG41-WFG48, the decision variables have k position related parameters and l distance related parameters, i.e., $n=k+l$, where k and l are respectively set to $2\times(m-1)$ and 20 as suggested in [54]; for MaF1-MaF7, n is set by $n=m+k-1$, where k is set to 10 for MaF1-MaF6 and to 20 for MaF7 as suggested in [59]; n=2 and n=5 are respectively used for MaF8-MaF9 and MaF13. Due to page limitations, the main characteristics of these test problems are summarized in Table A.I of the supplementary file.

In this paper, the well-known HV [63] and inverted generational distance (IGD) [64] are used as the performance indicators. Both HV and IGD are able to reflect the convergence and diversity of the final solution set produced by the algorithms. A larger HV value or a smaller IGD value indicate a better approximation to the true PF. When computing HV, a reference point dominated by the nadir point of the true PF is carefully specified for various problems. To ensure a fair comparison, the reference point of all test problems in our experiments are set as suggested in [65] and [66]. All objective values in the final solution set are firstly normalized by using $1.1\times(f_1^{nad}, f_2^{nad},..., f_m^{nad})$, where f_k^{nad} is the kth member of the nadir point in the true PF ($k = 1,2,...,m$), and then the reference point is set to the m-D point $(1.0,1.0,...,1.0)$. Here, the recently proposed walking fish group algorithm [67] is used to compute the exact HV values for problems with $m \le 10$, and the Monte Carlo simulation [13] using 10^7 sampling points is used to calculate HV for problems with m=15. To calculate IGD, a large set of points that are evenly sampled from the true PF is required. In particular, 5000, 10000, and 20000 points are uniformly sampled from the true PF to calculate the IGD values of 2-objective, 3-objective, and 5- to 15-objective problems, respectively. Due to page limitations, please refer to [13], [64] and [67] for details of computing HV and IGD.

B. Compared Algorithms and Parameters Settings

In this study, eight competitive MOEAs, i.e., NSGA-III [20], θ-DEA [69], VaEA [49], MaOEA/C [54], MOEA/D-LTD [70], PaRP/EA [56], MOEA/AD [33], and DDEA [57], are included for performance comparison. In NSGA-III and θ-DEA, N weight vectors evenly extracted on UH^1 are used in their optimization process, whereas in MOEA/D-LTD, the weight vectors and aggregation methods are adaptively set by learning the characteristics of the estimated PFs for various problems. Moreover, in VaEA, MaOEA/C and DDEA, solutions are treated as weight vectors. The angles between solutions are used to reflect their direction similarity as done in VaEA and MaOEA/C, while the distances of solutions' projections on UH^1 are employed in DDEA. Furthermore, two adversarial directions are considered in MOEA/AD by using two sets of weight vectors, and the direction similarity of solutions is adaptively computed by using two adversarial directions in PaRP/EA. The parameters settings of these MOEAs are provided in Table A. II of the supplementary file, as suggested in their references. Particularly, FDEA and all compared MOEAs use the same evolutionary operators, i.e., SBX and polynomial-based mutation.

The settings of population size for different numbers of objectives are listed in Table A. III of the supplementary file. For test problems with 2, 3, 5, 8, 10 and 15 objectives, the numbers of weight vectors are respectively set to 100, 120, 210, 240, 275 and 240, using the two-layer generation method with the simplex-lattice design factor H in [44]. According to [54], the population size in MaOEA/C should be set as a multiple of m. Thus, its population size on 10-objective test problems was set to 280. For other compared MOEAs, their population sizes are set the same as the number of weight vectors. All the compared MOEAs are run 30 times independently on each test problem. The mean HV and IGD values and their standard deviations (included in brackets after the mean HV or IGD results) from 30 runs are collected for comparison. All the compared MOEAs are terminated when a

TABLE I
PERFORMANCE OF EIGHT MOEAS ON 2- AND 3-OBJECTIVE WFG AND WFG4X PROBLEMS WITH IGD

Problems	m	NSGA-III	θ-DEA	VaEA	MaOEA/C	DDEA+NS	MOEA/D-LTD	PaRP/EA	FDEA
WFG1	2	8.950E-1(9.98E-2)−	8.550E-1(8.89E-2)−	8.831E-1(1.29E-1)−	9.256E-1(1.21E-1)−	**1.570E-1(4.12E-2)+**	1.760E-1(5.06E-2)+	8.964E-1(1.36E-1)−	7.625E-1(7.85E-2)
	3	9.389E-1(8.41E-2)−	9.006E-1(1.02E-1)−	8.324E-1(9.72E-2)−	1.104E+0(1.26E-1)−	2.805E-1(4.39E-2)+	**2.385E-1(2.91E-2)+**	9.194E-1(8.38E-2)−	7.225E-1(1.06E-1)
WFG2	2	1.262E-1(9.43E-2)−	1.198E-1(9.29E-2)~	1.464E-1(8.91E-2)−	1.303E-1(9.33E-2)−	**1.017E-1(2.15E-3)+**	1.264E-1(9.33E-2)−	1.278E-1(9.52E-2)−	1.181E-1(9.38E-2)
	3	2.290E-1(7.42E-2)−	2.463E-1(6.92E-2)−	2.730E-1(6.98E-2)−	2.957E-1(5.37E-2)−	2.443E-1(6.51E-2)−	2.865E-1(3.85E-2)−	2.232E-1(5.33E-2)−	**2.106E-1(6.37E-2)**
WFG3	2	2.135E-2(5.35E-3)−	2.024E-2(2.57E-3)−	1.998E-2(2.14E-3)−	1.622E-2(1.60E-3)~	1.704E-2(1.57E-3)−	1.833E-2(3.34E-3)−	2.099E-2(3.59E-3)−	**1.467E-2(8.84E-4)**
	3	1.041E-1(1.33E-2)−	1.076E-1(1.58E-2)−	1.605E-1(1.32E-2)−	7.061E-2(1.05E-2)~	7.098E-2(5.48E-3)~	8.597E-2(8.81E-3)−	7.650E-2(1.31E-2)−	**7.040E-2(4.80E-3)**
WFG41	2	1.471E-2(7.95E-4)+	1.464E-2(8.99E-4)+	1.706E-2(7.31E-4)−	1.578E-2(5.72E-4)~	**1.396E-2(3.18E-4)+**	1.593E-2(8.42E-4)~	1.706E-2(9.51E-4)−	1.582E-2(7.79E-4)
	3	1.923E-1(9.06E-4)~	**1.916E-1(7.45E-4)+**	2.088E-1(3.84E-3)~	2.049E-1(3.33E-3)~	1.967E-1(2.27E-3)+	2.087E-1(1.97E-3)~	2.036E-1(3.48E-3)~	2.055E-1(2.50E-3)
WFG42	2	2.710E-2(9.77E-4)−	3.044E-2(1.28E-3)−	7.666E-2(2.09E-2)−	3.922E-2(6.16E-3)−	3.089E-2(3.81E-3)−	2.808E-2(1.46E-3)−	2.882E-2(3.39E-3)−	**1.523E-2(5.42E-4)**
	3	1.864E-1(3.70E-3)−	1.884E-1(5.29E-3)−	2.212E-1(7.54E-3)−	2.478E-1(2.09E-2)−	1.944E-1(9.37E-3)−	2.086E-1(1.38E-2)−	2.052E-1(9.50E-3)−	**1.476E-1(2.01E-3)**
WFG43	2	3.216E-2(9.76E-3)−	3.341E-2(9.20E-3)−	3.310E-2(9.85E-3)−	3.014E-2(8.95E-3)−	4.685E-2(1.71E-2)−	**2.264E-2(2.49E-3)+**	3.090E-2(1.03E-2)−	2.416E-2(1.04E-2)
	3	3.075E-1(8.65E-3)−	3.051E-1(5.90E-3)−	2.767E-1(7.92E-3)~	2.901E-1(7.70E-3)−	3.110E-1(8.50E-3)−	3.960E-1(2.00E-2)−	2.771E-1(7.96E-3)~	**2.766E-1(6.96E-3)**
WFG44	2	1.734E-1(2.70E-2)−	2.019E-1(3.21E-2)−	4.269E-1(7.81E-2)−	2.011E-1(3.93E-2)−	1.727E-1(3.07E-2)−	2.228E-1(5.68E-2)−	1.638E-1(3.48E-2)−	**1.120E-1(2.59E-2)**
	3	1.811E-1(1.05E-2)−	2.396E-1(1.18E-2)−	2.843E-1(2.40E-2)−	2.803E-1(1.80E-2)−	2.190E-1(1.79E-2)−	1.923E-1(1.95E-2)−	1.880E-1(1.78E-2)−	**1.274E-1(2.00E-2)**
WFG45	2	**1.443E-2(5.61E-4)+**	1.446E-2(9.60E-4)+	1.692E-2(9.69E-4)−	1.543E-2(6.11E-4)~	1.492E-2(5.39E-4)~	1.526E-2(6.65E-4)~	1.688E-2(9.41E-4)−	1.565E-2(6.09E-4)
	3	2.033E-1(9.27E-4)~	2.020E-1(8.32E-4)~	2.043E-1(5.29E-3)~	2.056E-1(7.35E-3)~	2.060E-1(7.26E-3)~	2.212E-1(4.46E-3)−	**2.011E-1(5.37E-3)+**	2.037E-1(4.85E-3)
WFG46	2	1.450E-2(8.75E-4)−	1.411E-2(8.18E-4)−	1.632E-2(7.91E-4)−	1.395E-2(5.29E-4)~	1.476E-2(5.51E-4)−	1.628E-2(1.01E-3)−	1.768E-2(1.35E-3)−	**1.318E-2(3.74E-4)**
	3	1.513E-1(1.07E-3)~	**1.498E-1(9.51E-4)~**	1.698E-1(3.38E-3)−	1.624E-1(5.53E-3)−	1.568E-1(1.93E-3)~	1.645E-1(1.91E-3)−	1.602E-1(3.51E-3)~	1.505E-1(1.96E-3)
WFG47	2	1.223E-1(2.37E-1)−	1.024E-1(2.17E-1)−	8.331E-2(1.95E-1)−	2.218E-1(2.38E-1)−	9.971E-2(2.16E-1)−	1.247E-1(2.36E-1)−	1.841E-1(2.82E-1)−	**5.959E-2(1.59E-1)**
	3	2.863E-1(3.46E-1)−	3.332E-1(4.22E-1)−	3.705E-1(4.82E-1)−	**1.936E-1(5.83E-3)+**	3.320E-1(4.23E-1)−	3.617E-1(4.09E-1)−	4.146E-1(5.23E-1)−	2.366E-1(2.60E-1)
WFG48	2	2.316E-1(4.17E-1)−	1.302E-1(3.13E-1)−	1.711E-1(3.11E-1)−	2.631E-1(4.43E-1)−	1.958E-1(3.90E-1)−	1.699E-1(3.52E-1)−	2.258E-2(2.29E-3)~	**2.180E-2(2.62E-1)**
	3	1.921E-1(1.16E-2)~	2.820E-1(3.34E-1)−	6.609E-1(7.92E-1)−	3.499E-1(4.65E-1)−	4.689E-1(6.32E-1)−	3.074E-1(3.28E-1)−	2.713E-1(3.36E-1)−	**1.905E-1(1.12E-2)**
WFG9	2	**4.145E-2(3.80E-2)~**	5.511E-2(4.65E-2)−	4.336E-2(3.75E-2)~	4.967E-2(4.24E-2)−	6.192E-2(5.19E-2)−	6.690E-2(5.13E-2)−	5.592E-2(4.61E-2)−	4.181E-2(3.80E-2)
	3	2.147E-1(2.33E-2)−	2.155E-1(2.36E-2)−	2.205E-1(2.16E-2)−	2.248E-1(2.32E-2)−	2.272E-1(2.88E-2)−	2.495E-1(1.60E-2)−	2.130E-1(2.31E-2)~	**2.080E-1(1.88E-2)**
Best/All		2/24	2/24	0/24	1/24	3/24	2/24	1/24	**13/24**
+/−/~		2/17/5	3/18/3	0/21/3	1/16/7	5/15/4	3/18/3	1/18/5	——

predefined maximum number of generations $G_{\max}$ is reached. The values of $G_{\max}$ are set to 300, 500, 600, 800, 1000 and 1500, respectively for 2-, 3-, 5-, 8-, 10- and 15-objective test problems. Their maximum number of function evaluations (*MFE*) can be easily obtained by computing $MFE = N \times G_{\max}$.

C. Comparison Results on WFG and WFG4x Problems

In this experiment, FDEA was compared with respect to NSGA-III, θ-DEA, VaEA, MaOEA/C, MOEA/D-LTD, PaRP/EA and DDEA on WFG1-WFG3, WFG41-WFG48 and WFG9 with different numbers of objectives. Here, a Wilcoxon rank sum test with a 0.05 significance level and a Wilcoxon signed ranks test from the tool KEEL [71] are used to ensure a statistically sound conclusion, which can show the statistically significant differences on the performance results. In the following tables, the symbols "+", "–" and "~" indicate that the comparison results of the corresponding algorithm are significantly better than, worse than, and similar to FDEA on tackling each problem with different objectives, "+/−/~" collects the corresponding numbers of the above statistical results, and 'Avg. rank' indicates the average performance ranks of FDEA and other compared MOEAs by Friedman's test from KEEL.

1) IGD Results on Problems with 2- and 3-objective

The average IGD results of FDEA and its seven competitors on WFG1-WFG3, WFG41-WFG48 and WFG9 with 2- and 3-objective are presented in Table I. As shown in the second last row of Table I, FDEA obtains the best results in 13 out of 24 problems, while NSGA-III, θ-DEA, VaEA, MaOEA/C, DDEA, MOEA/D-LTD and PaRP/EA perform best in 2, 2, 0, 1, 3, 2 and 1 problems, respectively. According to Wilcoxon rank sum test, when compared to NSGA-III, θ-DEA, VaEA, MaOEA/C, DDEA, MOEA/D-LTD and PaRP/EA, FDEA is respectively worse in 2, 3, 0, 1, 5, 3, 1 out of 24 cases, while it is respectively better in 17, 18, 21, 16, 15, 18 and 18 cases, which validate the superiority of FDEA for tackling these 2- and 3-objective WFG and WFG4x problems. Particularly, FDEA outperforms all competitors on WFG3, WFG42, WFG44 and WFG9, whereas it performs worse than NSGA-III and θ-DEA on WFG41 and WFG45, and worse than DDEA and MOEA/D-LTD on WFG1.

To visually show the performace, the final solution sets with the median IGD values obtained by FDEA and its seven competitors on these considered WFG and WFG4x problems with 2 and 3 objectives are plotted in Figs. A1-A24, which are provided in the supplementary file due to page limitations. As observed from these figures, the test problems have different PF shapes. For WFG1 with a mixed and biased PF in Figs. A1-A2, all the algorithms perform poorly on convergence, while only DDEA and MOEA/D-LTD perform better in terms of distribution. Regarding WFG2 with a mixed and disconnected PF in Figs. A3-A4, FDEA obtains a solution set with the most even distribution on both 2- and 3-objective cases. On the 2-objective instance of WFG3 with a linear PF in Fig. A5 and on the 3-objective instance of WFG3 with an irregular PF in Fig. A6, all the agorithms except for NSGA-III can obtain an evenly distributed set of solutions for the 2-objective case, but they perform relatively poorly for the 3-objective case. Regarding WFG41 with a regular concave PF in Figs. A7-A8, the solutions obtained by NSGA-III and θ-DEA are distributed more evenly. For WFG9 with a multi-modal concave PF in Figs. A9-A10, FDEA finds the solution set with the best distribution. On WFG42 with a convex PF (in Fig. A11-A12) and WFG44 with an extremely convex PF (in Fig. A13-A14), only FDEA can generate the entire PF, while other algorithms tend to maintain solutions that concentrate on the central part of the PF. Regarding WFG43 with a sharply concave PF (in Figs. A15-A16), FDEA shows the best distribution of solutions on the 3-objective case, but it is outperformed by MOEA/D-LTD on the 2-objective case. On WFG45 (in Figs. A17-A18) and WFG46 (in Figs. A19-A20) respectively with simple mixed and linear PFs, FDEA and its competitors obtain a similar distribution of their

TABLE II
SUMMARY OF SIGNIFICANCE TEST BETWEEN FDEA AND SEVEN MOEAS ON WFG AND WFG4X PROBLEMS WITH HV

Comparisons	NSGA-III		θ-DEA		VaEA		MaOEA/C		DDEA+NS		MOEA/D-LTD		PaRP/EA		FDEA
on	+/−/∼	Avg. rank	+/−/∼	Avg. rank	+/−/∼	Avg. rank	+/−/∼	Avg. rank	+/−/∼	Avg. rank	+/−/∼	Avg. rank	+/−/∼	Avg. rank	Avg. rank
m = 2, 3	4/14/6	4.4792	3/14/7	4.125	0/16/8	6.4583	2/14/8	4.3125	7/11/6	3.6667	5/12/7	4.5208	1/14/9	5.7917	**2.6458**
m = 5	2/9/1	5.0	3/6/3	3.8333	0/12/0	7.5	3/7/2	3.4583	2/7/3	4.4583	**6/3/3**	**1.75**	0/11/1	7.0833	2.9167
m = 8	0/12/0	6.25	0/11/1	6.3333	0/12/0	5.75	5/6/1	2.5833	2/10/0	4.1667	2/8/2	2.9167	0/12/0	6.3333	**1.6667**
m = 10	0/9/3	4.5833	1/9/2	4.3333	0/11/1	6.8333	5/7/0	3.3333	2/9/1	5.1667	**4/4/4**	2.75	1/11/0	6.9167	**2.0833**
m = 15	2/10/0	6.25	1/11/0	5.75	1/9/2	4.375	1/9/2	3.125	3/8/1	4.0833	2/10/0	5.4167	1/9/2	4.9167	**2.0833**
All	8/54/10	5.1736	8/51/13	4.75	1/60/11	6.2292	16/43/13	3.6208	16/45/11	4.2014	**19/37/16**	3.5458	3/59/10	6.1389	**2.3403**

TABLE III
PERFORMANCE OF EIGHT MOEAS ON 3-OBJECTIVE MAF PROBLEMS WITH IGD

Problems	m	NSGA-III	θ-DEA	VaEA	MaOEA/C	MOEA/AD	DDEA+NS	PaRPEA	FDEA
MaF1	3	5.115E-2(1.29E-3)-	6.878E-2(5.08E-4)-	3.806E-2(5.15E-4)-	3.802E-2(5.044E-4)-	3.740E-2(3.67E-4)∼	3.747E-2(6.97E-4)∼	3.745E-2(2.03E-4)∼	**3.732E-2(4.45E-4)**
MaF2	3	3.050E-2(4.49E-4)-	3.156E-2(2.86E-4)-	2.683E-2(3.13E-4)-	2.798E-2(6.72E-4)-	3.743E-2(1.24E-3)-	2.588E-2(2.80E-4)-	2.699E-2(4.57E-4)-	**2.456E-2(6.48E-4)**
MaF3	3	6.322E-2(2.73E-2)-	4.706E-2(1.62E-3)-	5.090E-2(3.26E-3)-	1.832E-1(2.50E-2)-	1.120E-1(7.53E-2)-	1.659E-1(1.19E-1)-	5.340E-2(1.24E-2)-	**3.350E-2(3.51E-3)**
MaF4	3	2.971E-1(1.73E-2)-	3.209E-1(2.11E-2)-	2.900E-1(1.54E-2)-	5.001E-1(9.67E-2)-	2.937E-1(9.85E-2)-	3.472E-1(5.11E-2)-	2.721E-1(1.21E-2)∼	**2.720E-1(2.37E-2)**
MaF5	3	3.784E-1(8.52E-1)-	3.784E-1(8.52E-1)-	4.269E-1(8.74E-1)-	2.425E-1(6.93E-3)∼	2.772E-1(2.47E-2)-	3.496E-1(4.54E-1)-	**2.254E-1(3.43E-3)+**	2.300E-1(3.07E-3)
MaF6	3	1.186E-2(1.40E-3)-	2.804E-2(2.80E-3)-	4.717E-3(3.98E-4)-	4.947E-3(1.09E-3)-	**3.235E-2(3.33E-3)∼**	4.270E-3(2.05E-4)-	3.792E-3(1.65E-4)-	3.247E-3(3.32E-4)
MaF7	3	6.491E-2(2.10E-3)-	9.999E-2(7.16E-2)-	6.978E-2(5.47E-2)-	6.395E-2(5.43E-2)-	6.495E-2(5.43E-2)-	6.458E-2(5.60E-2)-	5.901E-2(1.69E-3)-	**5.096E-2(7.64E-3)**
MaF8	3	9.377E-2(7.00E-3)-	1.415E-1(4.78E-2)-	6.588E-2(2.94E-3)-	8.123E-2(6.91E-3)-	6.451E-2(3.18E-3)-	6.565E-2(5.80E-3)-	6.261E-2(1.16E-3)-	**6.141E-2(1.69E-3)**
MaF9	3	5.962E-2(9.23E-3)-	6.26E-2(1.21E-2)-	6.071E-2(2.54E-3)-	6.770E-2(4.96E-3)-	2.392E-1(1.46E-1)-	3.856E-1(7.29E-2)-	9.194E-1(8.38E-2)-	**5.569E-2(4.53E-4)**
MaF13	3	9.001E-2(1.29E-2)-	9.15E-2(1.90E-2)-	7.698E-2(7.96E-3)-	8.400E-2(1.15E-2)-	1.502E-1(2.66E-2)-	7.975E-2(1.01E-2)-	**2.699E-2(4.57E-4)+**	6.777E-2(3.65E-3)
Best/All		0/10	0/10	0/10	0/10	1/10	0/10	2/10	**7/10**
+/-/∼		0/10/0	0/10/0	0/10/0	0/9/1	0/8/2	0/9/1	2/6/2	——

solutions. For WFG47 (in Figs. A21-A22) and WFG48 (in Figs. A23-A24), in which their PFs are discontinuous with three segments, NSGA-III, VaEA and PaRP/EA only find the solutions with poor distribution around the first segment of 2-objective WFG47, MaOEA/C cannot find any solution around its last segment, whereas MOEA/D-LTD only gets a poorly distributed solution set on 3-objective WFG47. Moreover, only FDEA can obtain solutions around the first segment of 2-objective WFG48 and also shows the best distribution of solutions for its 3-objective case.

Based on the above analysis, NSGA-III and θ-DEA can only solve the WFG and WFG4x problems with regular PFs (like WFG41 and WFG46), as they use the fixed and evenly distributed weight vectors. MOEA/D-LTD can adaptively adjust the weight vectors and aggregation method by a learning model, which enhances the performance in some irregular problems (like WFG1 and WFG43), but it still faces some challenges on problems with discontinuous PFs, such as WFG2, WFG47 and WFG48, especially in their 3-objective cases. Moreover, although solutions are treated as weight vectors in MaOEA/C, VaEA and DDEA, they still fail to solve problems with convex PFs (like WFG42 and WFG44) or with a sharp convex segment on PFs (like WFG2, WFG47 and WFG48). Furthermore, PaRP/EA cannot deal with problems having discontinuous, extremely concave or convex PFs (like WFG43, WFG44, WFG47), even when using two adversarial directions. Considering our FDEA, as the fuzzy decomposition includes a fuzzy prediction to estimate the PF shape and a weight vector extraction to get N constrained subproblems, it can properly solve different problems by adaptively fitting their complex PF shapes. Based on the experimental studies on WFG and WFG4x problems, it is reasonable to conclude that FDEA achieves a superior performance on all of these selected problems except for WFG1.

2) HV Results on Problems with 2 to 15 objectives

Due to page limitations, the median IGD and HV of FDEA and its seven competitors on those WFG and WFG4x problems with 2 to 15 objectives are given in Table A. IV to Table A. VI of the supplementary file. Here, the statistical results using Wilcoxon rank sum test and Wilcoxon signed ranks test from KEEL [71] are summarized in Table II based on the HV results. From the separately statistical results with the value of m, FDEA is significantly better than its seven competitors on both WFG and WFG4x problems with multiple objectives (i.e., m = 2 or 3) and many objectives (i.e., m>3). Specifically, FDEA is only outperformed by MOEA/D-LTD on the 5-objective case, where MOEA/D-LTD is better than FDEA in 6 out of 12 problems and has the best performance rank (1.75), while FDEA is ranked with 2.9176. According to the statistical results for all cases in the last row of Table II, FDEA is advantageous on solving the WFG and WFG4x problems as it has the best rank (2.3403) in terms of HV.

D. Comparison of Results on the MaF Problems

As discussed above, FDEA is highly competitive or significantly superior to its competitors on the WFG and WFG4x problems with discontinuous, convex or sharp concave PFs. In this section, the performance of FDEA is further studied on solving the problems with other irregular PF shapes, e.g., the inverted and degenerated cases. Here, FDEA is compared with NSGA-III, θ-DEA , VaEA, MaOEA/C, MOEA/AD, PaRP/EA and DDEA on solving MaF1-MaF9 and MaF13 with different objectives $m \in \{3,5,8,10,15\}$. As MOEA/AD adopts two adversarial sets of weight vectors to get two solution sets correspondingly, only the solution set with the better IGD/HV performance is reported as its final solution set.

1) IGD Results on Problems with 3 objectives

The average IGD results of FDEA and its seven competitors on MaF1-MaF9 and MaF13 with m=3 are presented in Table III. As shown in the second last row of Table III, FDEA obtains the best results in 7 out of 10 problems, while NSGA-III, VaEA, θ-DEA , MaOEA/C, MOEA/AD, DDEA and PaRP-/EA perform respectively best in 0, 0, 0, 0, 1, 0 and 2 problems. According to Wilcoxon's rank sum test, FDEA is only worse than PaRP/EA on MaF5 and MaF13, and is better than or at least similar to other compared algorithms on the remaining cases. Thus, the advantages of FDEA on these 3-objective MaF problems are validated.

TABLE IV
SUMMARY OF SIGNIFICANCE TEST BETWEEN FDEA AND SEVEN MOEAS ON MAF1-MAF7 PROBLEMS WITH HV

Comparisons	NSGA-III		θ-DEA		VaEA		MaOEA/C		MOEA/AD		DDEA+NS		PaRP/EA		FDEA
on	+/−/~	Avg. rank	+/−/~	Avg. rank	+/−/~	Avg. rank	+/−/~	Avg. rank	+/−/~	Avg. rank	+/−/~	Avg. rank	+/−/~	Avg. rank	Avg. rank
$m = 3$	0/6/1	5.7143	0/6/1	5.2143	0/5/2	4.2145	0/4/3	4.7143	0/5/2	5.8571	0/6/1	6.0	**1/2/4**	2.7857	**1.5**
$m = 5$	1/6/0	4.7143	1/6/0	5.8571	0/6/1	4.7143	0/5/2	4.1429	0/5/2	5.0	0/4/3	5.0	1/4/2	4.7143	**1.8571**
$m = 8$	0/6/1	5.0	0/7/0	6.0	0/7/0	5.1429	**3/3/2**	3.0	0/7/0	4.7143	1/5/1	3.7143	0/7/0	7.1429	**1.2857**
$m = 10$	1/4/2	3.6429	2/5/0	4.8571	1/6/0	5.50	**3/3/2**	3.4286	2/5/0	4.7143	**3/2/2**	**3.2143**	0/6/1	7.2857	3.3571
$m = 15$	2/4/1	5.0	2/4/1	4.5714	2/4/1	4.7857	**3/2/2**	**3.5**	2/4/1	4.0	**4/2/1**	**3.0**	0/6/1	7.5	3.6429
All	4/26/5	4.8143	5/28/2	5.3	3/28/4	4.8714	10/17/11	3.7571	4/26/5	4.8571	8/19/8	4.2857	2/25/8	5.8857	**2.2286**

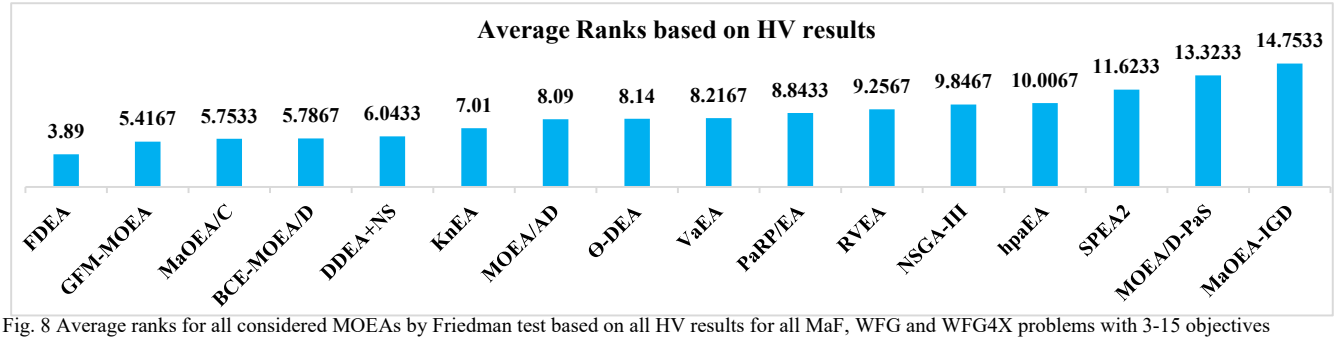


Fig. 8 Average ranks for all considered MOEAs by Friedman test based on all HV results for all MaF, WFG and WFG4X problems with 3-15 objectives

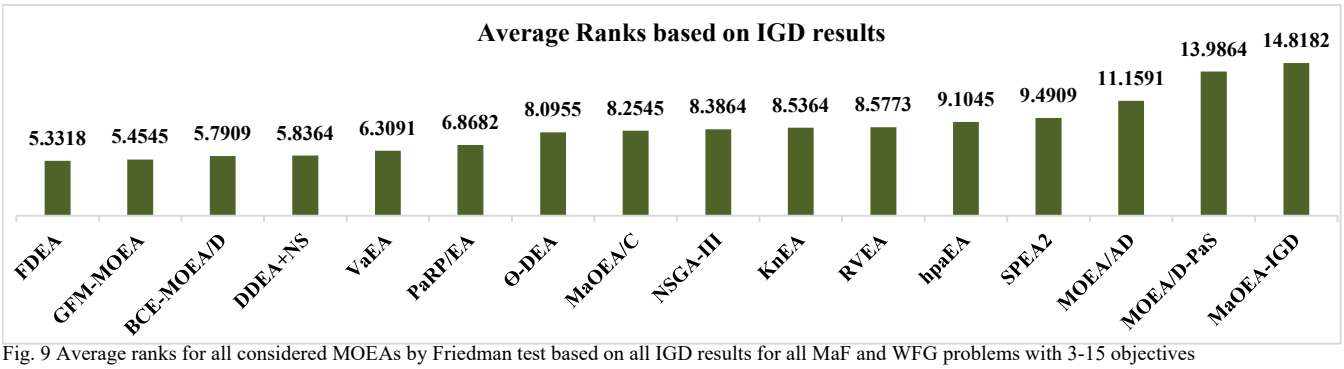


Fig. 9 Average ranks for all considered MOEAs by Friedman test based on all IGD results for all MaF and WFG problems with 3-15 objectives

In order to visually show their performace, the final solution sets with the median IGD values obtained by all the compared MOEAs on MaF1-MaF7 problems with 3 objectives are shown in Figs. A25-A31, which are provided in the supplementary file due to page limitations. From these figures, it can be observed that each MaF problem has a different and irregular PF shape. For MaF1 with an inverted linear PF (in Fig. A25), MOEA/AD and FDEA obtain the best solution sets in view of distribution, while NSGA-III and θ-DEA generate poorly distributed solutions. On MaF2 with a partially mild concave PF (in Fig. A26), FDEA obtains the best solution set in terms of convergence and distribution. Considering MaF3 with a convex PF (in Fig. A27), only FDEA can get an evenly distributed solution set, while the other competitors can only find solutions concentrating on the central part of the PF. Regarding MaF4 with an inverted concave PF (in Fig. A28), FDEA performs slightly better than the other algorithms, although they actually cannot get an evenly distributed solution set. For MaF5 with a badly scaled concave PF (in Fig. A29), FDEA only gets the second-best result, while PaRP/EA performs best. On MaF6 with a degenerated PF (in Fig. A30), NSGA-III, θ-DEA and MOEA/AD cannot find an evenly distributed solution set, while other algorithms can almost cover the whole PF completely. Regarding MaF7 with four discontinuous PF segments (in Fig. A31), FDEA performs best as it can find solutions spread well on all segments of the PF.

Through the above experimental studies on the MaF problems, some conclusions can be drawn as follows. As the fixed weight vectors in NSGA-III and θ-DEA cannot properly match the target problems' PFs, such as inverted PFs in MaF1 and MaF4, degenerated PFs in MaF6 and discontinuous PFs in MaF7, they perform poorly on these problems. Although two adversarial sets of weight vectors are considered in MOEA/AD, it still cannot guarantee an appropriate match for the weight vectors and PFs when facing problems with degenerated PFs (like MaF6) and discontinuous PF (like MaF7). For VaEA, MaOEA/C, PaRP/EA, DDEA and our FDEA that regard solutions as weight vectors, FDEA is the best to solve the MaF problems, as the used fuzzy prediction can estimate the PF shapes, which helps to run the weight vector extraction in fuzzy decomposition.

2) HV Results on Problems with 3 to 15 objectives

Due to page limitations, the median HV and IGD values of all the compared algorithms on MaF1-MaF13 with 3 to 15 objectives are given in Tables A.VII and A.VIII of the supplementary file, whereas the statistical test results on MaF1-MaF7 with HV are summarized in Table IV. Obviously, FDEA is significantly superior to its seven competitors on MaF1-MaF7 with different objectives. On case of $m = 3$, FDEA gets the best rank 1.5, and only PaRP/EA with the rank 2.7857 approximates to FDEA, while other competitors are all significantly worse than FDEA. For many-objective MaF problems (i.e., $m > 3$), FDEA is outperformed by DDEA in the 10-objective case as DDEA is better than FDEA in 3 out of 7

problems, and is outperformed by MaOEA/C and DDEA in the 15-objective case as MaOEA/C and DDEA are respectively better than FDEA in 3 and 4 out of 7 problems. When considering all cases, FDEA is the best one for solving MaF1-MaF7, as it has the best rank (2.2286) in terms of HV.

E. Comparison Results on More Algorithms

To further validate the performance of FDEA, another eight MOEAs including SPEA2 [10], MOEA/D-PaS [37], BCE-MOEA/D [6], RVEA [44], KnEA [5], hpaEA [53], MaOEA-IGD [11], and GFM-MOEA [75], are considered here for the experimental studies. For a fair comparison, the parameters settings of these MOEAs are provided in Table A. II of the supplementary file, as suggested in their references, and other parameters are set the same with that described in Section IV. A-B. Moreover, the HV and IGD results of these MOEAs on WFG1-WFG9, WFG41-WFG48 and MaF1-MaF13 test problems with 3 to 15 objectives are provided in Tables A. IX- A. XI of the supplementary file. Moreover, two overall average ranks of each MOEA considered in this paper are obtained by Friedman test on KEEL [71] respectively based on these IGD and HV results, which are plotted in Fig. 8 and Fig. 9. From these summarized results, FDEA gets the best ranks on both HV-based and IGD-based rankings, which validate its superior performance when compared to these MOEAs.

F. More Discussions about the Shared Weight Vector and the Predicted Values for p

Here, two variants of FDEA are designed to study the influence of the shared w, with one using a fixed $w = (1, 1, .., 1)$ and the other one without using the shared w for the subproblems. Moreover, four other variants of FDEA are designed to study the influence of p, where p is not fuzzily predicted but fixed as p=0.5, p=1.0 and p=2.0 respectively for first three of these variants, and p is obtained by (9) instead of (10) for the last variant. Furthermore, the observations of the predicted value of p on all considered MOPs during the evolutionary process are also studied. Due to page limitations, details for the comparison results of these six variants with FDEA and the observations of p are provided in the supplementary file.

V. Conclusions and Future Work

In this paper, a fuzzy decomposition based MOEA has been proposed for tackling various MOPs, i.e., FDEA. This algorithm can fuzzily decompose a MOP into a set of constrained subproblems. To do this, fuzzy decomposition is run by using the fuzzy prediction to estimate the PF shapes and employing weight vector extraction to select solutions. Please note that the fuzzy prediction will finally estimate a UH^p, which helps to define a more precise metric for computing the direction similarity between solutions in the weight vectors extraction. This way, the decomposed subproblems can well fit the target problem's PF. At last, only m corner solutions are selected to maintain diversity and other solutions are chosen for each of the remaining subproblems. In this case, the best convergence can be achieved by using the WS aggregated function and the shared weight vector from all the extracted weight vectors. When compared to eight competitive MOEAs (NSGA-III, θ-DEA, VaEA, MaOEA/C, DDEA, MOEA/D-LTD, MOEA/AD and PaRP/ EA, FDEA has shown to be better in most cases, especially on problems with irregular PFs.

In our future work, the prediction of the population's shape with more complicated models and the measurement of direction similarity between solutions will be further studied on more complicated MOPs. The application of FDEA on the real-world problems will be also conducted in our future work.

References

[1] Y. Tian, R. Chen, X. Zhang, Y. Su, Y. Jin, "A Strengthened Dominance Relation Considering Convergence and Diversity for Evolutionary Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 331-345, 2019.

[2] A. Ponsich, A. L. Jaimes, C. A. Coello Coello, "A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321-344, 2013.

[3] M. Gong, Z. Wang, Z. Zhu, L. Jiao, "A Similarity-Based Multiobjective Evolutionary Algorithm for Deployment Optimization of Near Space Communication System," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 878-897, 2017.

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

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

[6] M. Li, S. Yang, and X. Liu, "Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, 2016.

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

[8] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, Ma, USA: Kluwer Academic, 1999.

[9] 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, 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des., Optimisation Control.*, pp. 95–100, 2002.

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

[12] S. Jiang, J. Zhang, "A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm," *IEEE Trans. Cybernetics.,* vol. 45, no. 10, pp. 2202-2213, 2015.

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

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

[15] H. Sato, "Chain-reaction solution update in MOEA/D and its effects on multi- and many-objective optimization," *Soft Comput.,* vol. 20, pp. 3803-3820, 2016.

[16] 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.*, vol. 20, no. 2, pp. 180-198, 2016.

[17] H. Xu, W. Zeng, D. Zhang, X. Zeng, 'MOEA/HD: A Multiobjective Evolutionary Algorithm Based on Hierarchical Decomposition,' *IEEE Trans. Cybernetics.*, vol. 49, no. 2, pp. 517-526, 2019.

[18] J. Chen, J. Li, B. Xin, "DMOEA-εC: Decomposition-Based Multiobjective Evolutionary Algorithm With the ε-Constraint Framework," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 714-730, 2017.

[19] I. Das. J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

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

[21] K. Fang and C. Ma, "Orthogonal and uniform experimental design," *Science and Technology Press*, Beijing, 2001.

[22] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169-190, 2017.

[23] S. Jiang, Z. Cai, J. Zhang, and Y. Ong, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *Seventh International Conference on Natural Computation*, ICNC 2011, Shanghai, China, 26-28 July, 2011, pp. 1260–1264, 2011.

[24] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in hypervolume calculation for fair comparison and efficient search," in GECCO'17: Proc. of the *2017 Genetic and Evol. Comput. Conference*, pp. 585–592, 2017.

[25] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 5, pp. 3677–3688, 2012.

[26] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected Pareto fronts," *in Proc. Int. Conf. Evol. Multi Criterion Optim.*, pp. 127–140, 2015.

[27] M. Li, X. Yao, "What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multi-objective optimization," *Evolutionary Computation,* vol. 28, no. 2, pp. 227-253, 2020.

[28] M. Asafuddoula, H. K. Singh, T. Ray, "An Enhanced Decomposition-Based Evolutionary Algorithm With Adaptive Reference Vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2321-2334, 2018.

[29] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, "An Indicator Based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609-622, 2018.

[30] X. Cai, Z. Mei, Z. Fan, "A Decomposition-Based Many-Objective Evolutionary Algorithm With Two Types of Adjustments for Direction Vectors," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2335-2348, 2018.

[31] F. Q. Gu, Y. M. Cheung, "Self-organizing Map-based Weight Design for Decomposition-based Many-objective Evolutionary Algorithm," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211-225, 2018.

[32] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, C.L. P. Chen, "A Many-Objective Evolutionary Algorithm with Two Interacting Processes: Cascade Clustering and Reference Point Incremental Learning," *IEEE Trans. Evol. Compu.*, vol. 23, no. 4, pp. 572-586, 2019.

[33] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary manyobjective optimization based on adversarial decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 753-764, 2020.

[34] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, and L. Jiao, "On the use of two reference points in decomposition based multiobjective evolutionary algorithms," *Swarm and Evol. Comput.*, vol. 34, pp. 89–102, 2017.

[35] A. Zhou and Q. Zhang, 'Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms,' *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, 2016.

[36] K. Li, S. Kwong, Q. Zhang, K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, 2015.

[37] R. Wang, Q. Zhang, T. Zhang, "Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821-837, 2016.

[38] H. Ishibuchi, N. Akedo, and Y. Nojima, "A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems," in *Learning and Intelligent Optimization*. Heidelberg, Germany: Springer, pp. 231–246, 2013.

[39] P. C. Borges and M. P. Hansen, "A study of global convexity for a multiple objective travelling salesman problem," *in Essays and Surveys in Metaheuristics*. New York, NY, USA: Springer, 2002, pp. 129–150.

[40] L. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, "Constrained Subproblems in a Decompositon-Based Multiobjective evolutionary Algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, 2016.

[41] S. Liu, Q. Lin, K. Wong, L. Ma, C. A. Coello Coello, D. Gong, "A novel multi-objective evolutionary algorithm with dynamic decomposition strategy," *Swarm and Evol. Comput.*, vol. 48, pp. 182-200, 2019.

[42] R. Wang, Z. B. Zhou, and H. ishibuchi, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3-18, 2018.

[43] S. Jiang, S. Yang, "A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multi-objective and Many-objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329-346, 2017.

[44] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773-791, 2016.

[45] X. Cai, Z. X. Yang, Z. Fan, Q. F. Zhang, "Decomposition-Based-Sorting and Angle-Based-Selection for Evolutionary Multiobjective and Many-Objective Optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2824-2837, 2017.

[46] C. Liu, Q. Zhao, B. Yan, S. Elsayed, T. Ray, R. Sarker, "Adaptive Sorting-based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 247–257, 2019.

[47] M. Ming, R. Wang, Y. Zha, and T. Zhang, "Pareto adaptive penaltybased boundary intersection method for multi-objective optimization," *Inf. Sci.*, vol. 414, pp. 158–174, 2017.

[48] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296-313, 2018.

[49] Y. Xiang, Y. R. Zhou, M. Q. Li, "A Vector Angle based Evolutionary Algorithm for Unconstrained Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131-152, 2017.

[50] Z. N. He, G. G. Yen, "Many-Objective Evolutionary Algorithm Based on Coordinated Selection Strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220-233, 2017.

[51] H. Liu, L. Chen, Q. Zhang, K. Deb, "Adaptively Allocating Search Effort in Challenging Many-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 433-448, 2018.

[52] Y. Zhou, Y. Xiang, Z. Chen, J. He, J. Wang, "A Scalar Projection and Angle-Based Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2073-2084, 2019.

[53] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, L. Wang, "Hyperplane Assisted Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3367-3380, 2020.

[54] Q. Lin, S. Liu, et al., "A Clustering-based Evolutionary Algorithm for Many-objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391-405, 2019.

[55] H. Bai, J. Zheng, G. Yu, S. Yang, J. Zou, "A Pareto-based many-objective evolutionary algorithm using space partitioning selection and angle-based truncation," *Inf. Sci.*, vol. 478, pp. 186-207, 2019.

[56] Y. Xiang, Y. Zhou, X. Yang, H. Huang, "A Many-objective Evolutionary Algorithm With Pareto-adaptive Reference Points," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 99-113, 2020.

[57] X. He, Y. Zhou, Z. Chen, Q. Zhang, "Evolutionary Many-objective Optimization based on Dynamical Decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 361-375, 2019.

[58] J. Cheng, G. Yen, G. Zhang, "A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 592-605, 2015.

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

[60] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, pp. 115–148, Apr. 1995.

[61] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[62] S. Huband, L. Barone, R. while, and P. Hingston, "A scalable multi-objective test problem tookit," in *Proc.3rd Conf. Evol .Multi-Criterion Optimiz.*, *Lecture Notes in Computer Science*, vol. 3410, pp. 280–295, 2005.

[63] 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, 1999.

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

[65] 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, pp. 87–102, 2009.

[66] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Manyobjective test problems to visually examine the behavior of multiobjective evolution in a decision space," in Proc. *Int. Conf. Parallel Prob. Solv. Nat., Krakow, Poland*, pp. 91–100, 2010.

[67] L. While, L. Bradstreet, L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, 2012.

[68] Y. Liu, H. Ishibuchi, N. Masuyama, Y. Nojima, "Adapting Reference Vectors and Scalarizing Functions by Growing Neural Gas to Handle Irregular Pareto Fronts," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 439-453, 2020.

[69] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16-37, February 2016.

[70] M. Wu, K. Li, S. Kwong, Q. Zhang, J. Zhang, "Learning to Decompose: A Paradigm for Decomposition-Based Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376-390, 2019.

[71] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrel, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Comput*., vol. 13, pp. 307–318, 2009.

[72] H. Wang, X. Yao, "Corner sort for Pareto-Based Many-Objective Optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92-102, 2014.

[73] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina, "Pareto-adaptive -dominance," *Evol. Comput.*, vol. 15, no. 4, pp. 493–517, 2007.

[74] S. Z. Martínez, V. A. S. Hernández, H. Aguirre, K. Tanaka, and C. A. Coello Coello, "Using a family of curves to approximate the Pareto front of a multi-objective optimization problem," in *Proc. 13th Int. Conf. Parallel Problem Solving Nat.*, 2014, pp. 682–691.

[75] Y. Tian, X. Zhang, R. Cheng, Y. Jin, "Guiding Evolutionary Multi-objective Optimization With Generic Front Modeling," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1106-1119, 2020.

[76] H. Xu, W. Zeng, X. Zeng, G. G. Yen, "An Evolutionary Algorithm Based on Minkowski Distance for Many-Objective Optimization," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3968-3979, 2019.
[77] H. Xu, W. Zeng, X. Zeng, G. G. Yen, "Polar-Metric-Based Evolutionary Algorithm," *IEEE Trans. Cybern.*, doi: 10.1109/TCYB.2020.2965230, in press, 2020.

Songbai Liu received the B.S. degree from Changsha University and the M.S. degree from Shenzhen University, China, in 2012 and 2018, respectively. He worked for ShenZhen TVT Digital Technology Co., Ltd as a software QA engineer from 2013 to 2015, and he worked for Shenzhen University as a research assistance from 2018 to 2019.

He is currently a PhD student in Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include nature-inspired computation, evolutionary many-objective optimization, and machine learning.

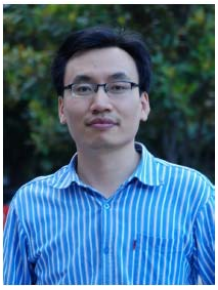
Qiuzhen Lin received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over twenty research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, and dynamic system.

Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) in electronics and electrical engineering and the Ph.D. degree in evolutionary computation and control systems from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Prof. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, was the Editor-in-Chief of the IEEE Computational Intelligence Magazine from 2010 to 2013, and currently serves as the Editorial Board Member of over ten journals. He was an Elected Member of IEEE CIS AdCom from 2017 to 2019.



Maoguo Gong (M'07–SM'14) received the B.S. and Ph.D. degrees in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively. Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and a Full Professor, respectively, both with exceptive admission. His current research interests include computational intelligence with applications to optimization, learning, data mining, and image understanding.

Dr. Gong was a recipient of the Prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is the Vice Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing, an Executive Committee Member of the Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He is also an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Carlos A. Coello Coello (M'98-SM'04-F'11) received Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996.

He is a Professor (CINVESTAV-3F Researcher) with the Department of Computer Science of CINVESTAV-IPN, Mexico City, México. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His publications currently report over 48,900 citations in Google Scholar (his h-index is 80). His research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Dr. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of *Exact Sciences*, the 2013 IEEE Kiyo Tomiyasu Award and the 2012 National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences. He is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves on the editorial board of 12 other international journals. He is a member of the Association for Computing Machinery, Sigma Xi, and the Mexican Academy of Science