An Effective Ensemble Framework for Multi-Objective Optimization

Wenjun Wang, Shaoqiang Yang, Qiuzhen Lin, Qingfu Zhang, *Fellow*, *IEEE*, Ka-Chun Wong, Carlos A. Coello Coello, *Fellow, IEEE*, and Jianyong Chen, *Member*, *IEEE*

***Abstract*—This paper proposes an effective ensemble framework for tackling multi-objective optimization problems, by combining the advantages of various evolutionary operators and selection criteria that are run on multiple populations. A simple ensemble algorithm is realized as a prototype to demonstrate our proposed framework. Two mechanisms, namely competition and cooperation, are employed to drive the running of the ensembles. Competition is designed by adaptively running different evolutionary operators on multiple populations. The operator that better fits the problem's characteristics will receive more computational resources, being rewarded by a decomposition-based credit assignment strategy. Cooperation is achieved by a cooperative selection of the offspring generated by different populations. In this way, the promising offspring from one population have chances to migrate into the other populations to enhance their convergence or diversity. Moreover, the population update information is further exploited to build an evolutionary potentiality model, which is used to guide the evolutionary process. Our experimental results show the superior performance of our proposed ensemble algorithms in solving most cases of a set of thirty-one test problems, which corroborates the advantages of our ensemble framework.**

***Index Terms*—ensemble framework, multi-objective optimization, competitive evolution, cooperative selection.**

I. Introduction

In the last decade, a number of multi-objective evolutionary algorithms (MOEAs) have shown remarkable performance in tackling various kinds of multi-objective optimization problems (MOPs) [1]-[3]. An unconstrained MOP can be defined as

$$\min_{\mathbf{x}\in\Omega} F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^T, \quad (1)$$

where $\mathbf{x}= (x_1, x_2,\ldots, x_n)$ is an n-dimensional decision vector bounded in the search space Ω, and m is the number of objectives [2]. Due to the inherent conflicts among the objectives, no single solution is optimal for all the objectives in (1). Instead, a set of optimal solutions termed Pareto-optimal set (PS) exists. The mapping of PS in the objective space is termed Pareto-optimal front (PF). As some MOPs in real-world applications often exhibit complex characteristics (e.g., nonlinearity, discontinuity, multimodality, degeneration, and a high dimensional decision space [4]), traditional deterministic methods are not so effective for these challenging MOPs [2]. For such problems, state-of-the-art MOEAs (e.g., SPEA2 [5], NSGA-II [6], MOEA/D [7], and IBEA [8]) are more advantageous and robust. Each of these state-of-the-art MOEAs shows certain effectiveness in tackling some kinds of MOPs. However, no MOEA with specific parameter setting, evolutionary operator, and selection criterion, can be the best performer for all types of MOPs. Thus, in order to achieve a better overall performance for a diverse range of MOPs, the use of ensemble approaches for MOEAs has been proposed. It is a natural and intuitive idea to combine the advantages of different parameter settings, evolutionary operators or selection criteria in MOEAs [3]. Such research efforts include the use of ensembles of heuristics [9]-[10], neighborhood sizes [11], niching methods [12], constraint handling techniques [13], and other related methods [14]-[19].

Ensemble of evolutionary operators. Different evolutionary operators show some advantages when tackling certain types of MOPs due to their different abilities on exploitation and exploration [3], [15]-[16]. For example, polynomial-based mutation (PM) [2] is good at exploitation (searching in a local area), while simulated binary crossover (SBX) [17], differential evolution (DE) [18], [41], and the estimation of distribution algorithm (EDA) [20] provide different search patterns for exploration. Thus, a number of research studies have been conducted to find a better ensemble of them, such as adaptive memetic computing (AMC) [21], ensembles of multiple DEs [10], [22], different mutation strategies [23]-[24], and various hyper-heuristic methods [9]. These approaches are designed by solving two basic issues: which operators should be included in the ensemble and how to run them. Regarding the first issue, evolutionary operators with distinct search patterns are widely used to compose an ensemble [25]-[26]. Otherwise, all the potentially optimal ensembles are tested in order and then the best of them is selected [9]. To tackle the second issue, many competition strategies [27]-[30] have been proposed to adaptively run evolutionary operators according to their performances. In this way, at different phases of the evolutionary process [31], various search mechanisms [32] can be flexibly used to tackle different types of MOPs.

Manuscript received XX. XX. XXXX; revised XX. XX. XXXX; accepted XX. XX. 2018. This work was supported by the National Natural Science Foundation of China under Grants 61876110, 61876163, and 61836005, a grant from ANR/RGC Joint Research Scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and France National Research Agency (Project No. A-CityU101/16), the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, and CONACyT grant no. 221551.

W.J. Wang, S.Q. Yang, Q.Z. Lin, and J.Y. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail of Qiuzhen Lin: qiuzhlin@szu.edu.cn).

Q.F. Zhang and K.C. Wong are with the Department of Computer Science, City University of Hong Kong, Hong Kong.

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

Ensemble of selection criteria. Selection criteria evaluate the solutions' quality in terms of both convergence (i.e., distance to the true PF) and diversity (i.e., distribution along the approximate PF). Ensemble approaches of different selection criteria have been widely studied in [33]-[38]. Superior solutions will be reserved according to the employed selection criterion, aiming to balance convergence and diversity for the population. As shown in BCE [38], when solving certain MOPs requiring high convergence pressure, a non-Pareto-based selection criterion should be preferred. Otherwise, a Pareto-based selection criterion with a diversity maintenance strategy is more appropriate. In [34], two archives with different selection criteria respectively focus on convergence and diversity. Similarly, in Two_Arch2 [35], the advantages of indicator and Pareto-based selection criteria are combined, with a new L_p-norm-based diversity maintenance scheme designed for many-objective optimization problems (i.e., MOPs with more than 3 objectives). In EAG [36], Pareto-based sorting and a decomposition-based strategy have been found to complement each other in environmental selection. In D^{2}MOPSO [37], Pareto-based ranking is employed to build an archive with the particle leaders, aiming to speed up convergence. Moreover, a decomposition-based method is exploited to update the particle's movement in order to provide a good coverage to the approximate PF.

Inspired by the above ensemble methods, a simple ensemble algorithm and its generalized ensemble framework (EF) are presented to tackle various MOPs with superior performance. The main contributions of this work are listed as follows:

1) A simple and effective EF is designed to solve MOPs, by running different evolutionary operators and selection criteria within multiple populations. Our proposed EF shows high scalability for embedding more than two populations with different search patterns. Each population competes to produce offspring using its own evolutionary operator and cooperates to select offspring using its exclusive criterion.

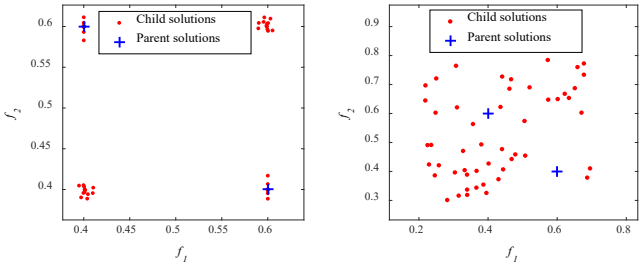
2) The competition among populations is realized by running different evolutionary operators, as awarded by the decomposition-based credit assignment strategy. These assigned credits decide the ratio of individuals from each population to undergo the specific evolutionary operators. This way, the performance of the evolutionary operators in each population can be easily quantified to guide the resource allocation in next generation.

The rest of this paper is organized as follows. Section II analyzes the competition and cooperation mechanisms in existing ensemble approaches. A simple ensemble algorithm is given in Section III, and then it is extended to a generalized EF in Section IV. Simulations are presented in Section V to study the effectiveness of our algorithm. Section VI gives more discussions on the credit assignment approaches, the matching strategies of populations and operators, and an extended instance with triple populations. Finally, our conclusions and some possible paths for future work are provided in Section VII.

II. Related Ensemble Approaches and Motivations

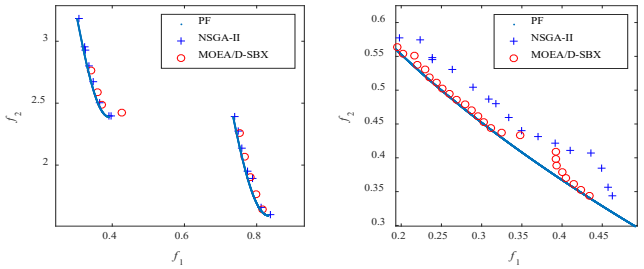
A. Competition on Evolutionary Operators

Competition mechanisms are usually designed to drive the running of evolutionary operators, as they may show different search patterns. An experiment was conducted in Fig. 1 to show the search behaviors of SBX and DE. Two parents $\mathbf{x}^1$ (0.4, 0.6) and $\mathbf{x}^2$ (0.6, 0.4) were used to run SBX [18] and a DE variant (DE/rand/1/bin) [41], where $\mathbf{x}^3$ for DE/rand/1/bin is randomly sampled in [0, 1]. In Fig. 1, 50 offspring solutions generated by each operator are plotted in decision space, with the parameters settings from [39]-[40]. As shown, the solutions from SBX are centralized around four corner points (two parents and two crossed points), while those from DE/rand/1/bin are distributed more evenly, as they are sampled by the joint probability distribution of $\mathbf{x}^2$ and $\mathbf{x}^3$ [41]. Therefore, some ensemble methods composed by SBX and DE, such as the hybrid recombination operator (DEI) [24], adaptive hybrid crossover (AHX) [24], and the hybridization of SBX and DE [26], often show a promising performance.



(a) SBX (b) DE/rand/1/bin

Fig. 1 Different search patterns of SBX and DE/rand/1/bin



(a) Local PF on WFG2 (b) Local PF on UF2

Fig. 2 Comparison between Pareto and decomposition-based selection criteria

B. Cooperation on Selection Criteria

Any selection criterion may show its own weakness in tackling various MOPs with complicated PFs. Thus, the cooperation in the ensemble of selection criteria is another important mechanism, which aims to complement one's potential weakness by the strength of the others. Pareto-based selection criteria may suffer from premature convergence [8], while decomposition and indicator-based selection criteria may get a poor distribution of solutions especially in tackling highly irregular MOPs [38]. An experiment that we performed is shown in Fig. 2, where a decomposition-based selection criterion in MOEA/D-SBX [7] and a Pareto-based one in NSGA-II [6] were used to solve the test problems (i.e., WFG2 [42] and UF2 [43]-[44]). In this experiment, SBX with the original parameters settings in [6], [40] was used. The approximate PFs of WFG2 and UF2 were plotted in Fig. 2 to show the weaknesses of diverse selection criteria.

In Fig. 2(a), the optimal solutions of MOEA/D-SBX that are associated to some weight vectors failed to cover the entire PF,

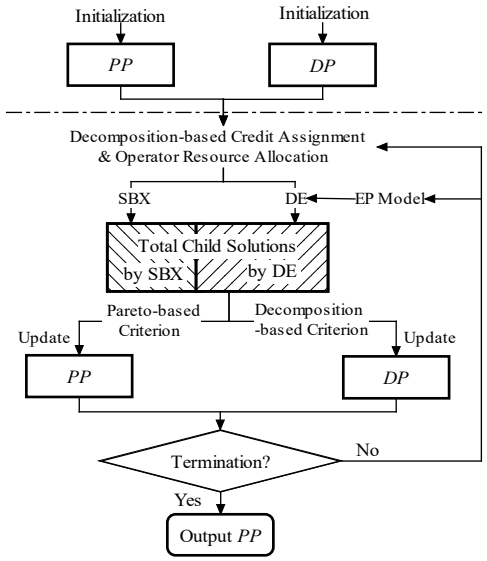


Fig. 3 Main flowchart of EF-PD

as a number of solutions are assigned to optimize the discontinuous components of WFG2. Thus, the decomposition-based selection criterion with uniform weight vectors is not so good at tackling the problems with discontinuous and irregular PFs. On the other hand, in Fig. 2(b), a number of final solutions found by NSGA-II are still far away from the PF, which indicates that the Pareto-based selection criterion is unable to provide strong convergence pressure on UF2.

C. Competitive and Cooperative Framework

Based on the above observations and discussions, it is found that most ensemble approaches emphasize either the competition of multiple evolutionary operators (e.g., [28], ACOS [29], and AMC [21]), or the cooperation of multiple selection criteria (e.g., MOMAD [45], EAG [36], NSGA-III [46] and MOEA/DD [47]). However, few studies have been carried out to combine competition and cooperation during one evolutionary process. A competition-cooperation framework was designed in BCE [38], which updates the two employed populations respectively using Pareto-based and non-Pareto-based criteria. However, it cannot be further extended with multiple co-evolved populations. In NSGA-III-HVOA [48], a competition-cooperation mechanism was designed by using a random strategy to choose its evolutionary operators. In its enhanced version (NSGA-III-AP) [49], this random strategy incorporates a probability-based adaptive operator selection method to reward the operator with the best performance. However, the algorithms in [48]-[49] only perform competition and cooperation in a single population. Therefore, these algorithms cannot be easily extended when more promising operators or selection criteria are available. Inspired and motivated by all the above studies, a more generalized EF consisting of multiple populations is proposed here to run diverse evolutionary operators and selection criteria. Our proposed EF is suitable for tackling various MOPs, using a competition of different evolutionary operators and cooperation of diverse selection criteria.

Algorithm 1: The complete EF-PD

1 **Initialization**;// (**Alg. 2**)
2 $ev = 0$, $g = 0$, $F_{DE} = 10^{-6}$, $F_{SBX} = 10^{-6}$, $S_{DE} = \{\}$, $S_{SBX} = \{\}$;
3 **while** $ev <$ *max-ev*
4 obtain N_{DE}^g and N_{SBX}^g respectively based on F_{DE} and F_{SBX};
5 evolve and evaluate N_{DE}^g solutions of *DP* by DE and the EP model to get S_{DE};
6 evolve and evaluate N_{SBX}^g solutions of *PP* by SBX to get S_{SBX};
7 [F_{DE}, F_{SBX}, *DP*, *PP*]=**Co_Selection**(S_{DE}, S_{SBX}, *DP*, *PP*);// (**Alg. 3**)
8 $ev = ev + N_{DE}^g + N_{SBX}^g$; g++;
9 **if** *mod*(g, 10) == 0
10 update the EP model;
11 **end if**
12 **end while**
13 **Output** *PP*;

III. A Simple Ensemble Algorithm

In this section, a simple algorithm based on the idea of **E**nsemble **F**ramework is introduced by employing **P**areto and **D**ecomposition-based populations, termed EF-PD. Existing studies have revealed that the search ability of DE is stronger than that of SBX under the decomposition framework when solving some complicated MOPs [50], while the search ability of SBX can be enhanced by selecting non-dominated solutions as elites [6], [52]. Thus, DE and SBX are respectively used to evolve the individuals in Pareto and decomposition-based populations. Fig. 3 shows the main flowchart of EF-PD, where *PP* indicates the population evolved by SBX and selected by a Pareto-based criterion, and *DP* denotes the population evolved by DE and selected by a decomposition-based criterion. The main mechanism of EF-PD is to run a competition on different evolutionary operators and a cooperation on diverse selection criteria.

To give an overview of EF-PD, its complete pseudo-code is provided in **Algorithm 1**. In Line 1, the **initialization** of the population is executed as presented in Section III.A. In Line 2, some algorithmic parameters are set, such as *ev* and *g* (the counters of the function evaluations and the generations, respectively), F_{DE} and F_{SBX} (the accumulated enhancements achieved by DE and SBX, respectively), S_{DE} and S_{SBX} (the offspring sets produced by DE and SBX, respectively). Please note that the initial values of F_{DE} and F_{SBX} are both set to 10^{-6} as there is no prior knowledge about their relative performance. After that, EF-PD runs the main evolutionary loop in Lines 3-12, until the maximum number of function evaluations (*max-ev*) is reached. The competition of SBX and DE is realized in Line 4, where N_{DE}^g and N_{SBX}^g indicate the number of executions of DE and SBX at generation *g*, respectively. Then, SBX and DE are run respectively in Lines 5-6 as introduced in Section III.B and the objectives of their offspring are evaluated. In Line 7, the cooperation between the Pareto and decomposition-based selection criteria is performed as presented in Section III.C. The counters *ev* and *g* are updated in Line 8. To further facilitate the cooperation, the proposed evolutionary potentiality (EP) model is updated at every 10 generations in Lines 9-11, as presented in Section III.D. When EF-PD terminates, the solutions in *PP* are reported as the final result.

Algorithm 2: Initialization

1 DP={}, PP={};
2 **for** i =1 to N
3 randomly generate an individual $\mathbf{x}^i$;
4 evaluate the objectives of $\mathbf{x}^i$;
5 add $\mathbf{x}^i$ to DP;
6 **end for**
7 initialize N weight vectors $\lambda^1,...,\lambda^N$;
8 **for** i=1 to N
9 $B^i=\{i_1,...,i_T\}$;// $\lambda^1,...,\lambda^N$ are the T closest weight vectors to λ^i
10 **end for**
11 initialize $\mathbf{z}^*$ by $\mathbf{z}_j^*=\min\{f_j(\mathbf{x})\mid \mathbf{x}\in DP\}$ for all $j\in\{1,...,m\}$;
12 keep the non-dominated solutions of DP in PP;

A. Population Initialization

The pseudo-code of **initialization** is given in **Algorithm 2**. First, DP and PP are initialized as an empty set. In Lines 2-6, $DP=\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ is randomly sampled from Ω by using a uniform distribution, where N is the population size. In order to decompose a MOP into a set of subproblems, the weight vectors set $W=\{\lambda^1,\lambda^2,...,\lambda^N\}$ is generated by a preset integer H and each dimensional value in λ^i (i=1, 2, … , N) is taken from {0/H, 1/H, ..., H/H} with the constraint $\sum_{j=1}^{m}\lambda_j^i=1$ (m is the number of objectives) [50]. This way, the weight vectors are uniformly distributed in objective space, and the number of weight vectors is $N=C_{H+m-1}^{m-1}$, requiring the population size to fit this equation. Then, the fitness value of solution $\mathbf{x}$ can be assigned by the Tchebycheff (TCH) approach, as follows:

$$g^{tch}(\mathbf{x}\mid\lambda,\mathbf{z}^*)=\max_{1\le j\le m}\left\{\lambda_j\times|\,f_j(\mathbf{x})-\mathbf{z}_j^*\,|\right\}, \tag{2}$$

where $f_j(\mathbf{x})$ stands for the j-th objective value. Each solution can be associated to a subproblem by

$$\lambda^*=\arg\ \underset{i=\{1,...,N\}}{\text{minimize}}\ g^{tch}(\mathbf{x}\mid\lambda^i,\mathbf{z}^*), \tag{3}$$

where $\mathbf{z}^*=(z_1^*,z_2^*,...,z_m^*)^T$ is the ideal vector for m objectives, which is approximated by the minimum value of each objective in the current population, i.e., for all $j\in\{1,...,m\}$

$$z_j^*=\min\{f_j(\mathbf{x})\mid\mathbf{x}\in DP\}. \tag{4}$$

Then, the neighbors of subproblem λ^i are determined in Lines 8-10. For each weight vector λ^i, let $B^i=\{i_1,i_2,...,i_T\}$ denote the neighborhood set of λ^i, where $\lambda^{i_1},...,\lambda^{i_T}$ are the T ($1\le T\le N$) closest weight vectors to λ^i according to the Euclidean distances between each pair of weight vectors. In Line 12, PP is initialized with the non-dominated solutions in DP, which provides parents of a higher quality for applying SBX. Please note that the number of individuals in PP may reach the population size N after running for several generations, depending on the number of non-dominated solutions.

B. Competitive Evolution of SBX and DE

Due to limited computational resources, the total number of executions of the operators in one generation is set to N in EF-PD. Both SBX and DE are adaptively run according to their credits (F_{SBX} and F_{DE}) as awarded by the average fitness improvement (AFI) at generation g, as follows:

$$\begin{cases} F_{\text{SBX}}=(\sum\Delta f^{i_1})/N_{\text{SBX}}^g & i_1\in I_{\text{SBX}}^g \\ F_{\text{DE}}\ =(\sum\Delta f^{i_2})/N_{\text{DE}}^g & i_2\in I_{\text{DE}}^g \end{cases}, \tag{5}$$

where I_{SBX}^g and I_{DE}^g denote the index sets of subproblems enhanced by SBX and DE, respectively. N_{SBX}^g and N_{DE}^g are the number of executions of SBX and DE, respectively at generation g. i_1 and i_2 stand for the indexes of subproblems in I_{SBX}^g and I_{DE}^g, respectively. Δf^i indicates the enhancement of the i-th subproblem under the TCH decomposition function, as follows:

$$\Delta f^i(\mathbf{x},\mathbf{y}\mid\lambda^i,\mathbf{z}^*)=g^{tch}(\mathbf{x}\mid\lambda^i,\mathbf{z}^*)-g^{tch}(\mathbf{y}\mid\lambda^i,\mathbf{z}^*), \tag{6}$$

where $\Delta f^i(\mathbf{x},\mathbf{y}\mid\lambda^i,\mathbf{z}^*)$ is the enhancement brought by the new solution $\mathbf{y}$ associated to the i-th subproblem over the original associated solution $\mathbf{x}$. After that, the normalized credits (F_{S} and F_{D}) can be obtained by

$$\begin{cases} F_{\text{S}}=F_{\text{SBX}}/(F_{\text{SBX}}+F_{\text{DE}}) \\ F_{\text{D}}=F_{\text{DE}}/\ (F_{\text{SBX}}+F_{\text{DE}}) \end{cases}, \tag{7}$$

where $0\le F_{\text{S}}, F_{\text{D}}\le 1$. Thus, based on F_{S} and F_{D}, the number of executions of DE and SBX at generation g+1 (N_{SBX}^{g+1} and N_{DE}^{g+1}) can be calculated by

$$\begin{cases} N_{\text{DE}}^{g+1}=\mathbf{max}(\mathbf{min}(\lfloor F_{\text{D}}\times N\rfloor,N-1),1) \\ N_{\text{SBX}}^{g+1}=N-N_{\text{DE}}^{g+1} \end{cases}, \tag{8}$$

where the functions **max**() and **min**() respectively return the maximum and minimum values. Here, the values of N_{SBX}^{g+1} and N_{DE}^{g+1} are manually set such that their minimal allowable value is 1, so as to keep each operator running at least once in each generation. Other minimal allowable numbers, such as $1\%\cdot N$ to $5\%\cdot N$, can be also specified by the user in (8). Due to page limitations, the pseudo-codes of SBX and DE are respectively presented in **Algorithm S-1** and **Algorithm S-2**, in Section S-I of the supplementary file.

Algorithm 3: Co_Selection (S_{DE}, S_{SBX}, DP, PP)

1 SI=RandomPermutation($S_{\text{DE}}\cup S_{\text{SBX}}$), F_{SBX}=0, F_{DE}=0;
2 **for** i = 1 to $|SI|$
3 **if DP_Selection**(SI_i, DP) == true
4 **if** SI_i is generated by SBX
5 $F_{\text{SBX}}=F_{\text{SBX}}+\Delta f^i$;
6 **else** $F_{\text{DE}}=F_{\text{DE}}+\Delta f^i$;
7 **end if**
8 $EP^{\text{subproblemof}(SI_i)}=1.0$;
9 **end if**
10 **if PP_Selection**(SI_i, PP) == true
11 **if** SI_i is generated by DE
12 $EP^{\text{subproblemof}(SI_i)}=1.0$;
13 **end if**
14 **end if**
15 **end for**
16 **return** F_{DE}, F_{SBX}, DP, PP;

C. Cooperative Selection by Pareto-based and Decomposition-based Criteria

To clarify the procedure of cooperative selection, **Algorithm 3** is provided with the inputs S_{DE} (the offspring set generated by DE), S_{SBX} (the offspring set generated by SBX), DP and PP. S_{SBX} and S_{DE} are first combined as a union set SI that is permuted randomly in Line 1. Please note that the sequence of offspring in SI can be either ordered or randomized, since our experiments indicate that the sequence order of offspring didn't significantly affect the final results. **DP_Selection** (SI_i, DP) and **PP_Selection** (SI_i, PP) respectively denote the selection mechanisms associated to the use of the decomposition-based and Pareto-based approaches introduced in [6] and [48], where SI_i indicates the i-th offspring from SI. Due to page limitations, their pseudo-codes are given in **Algorithm S-3** and **Algorithm**

S-4 of the supplementary file, respectively. In Lines 2-15 of **Algorithm 3**, each SI_i will be checked to see whether it can update *DP* or *PP*. If SI_i can update any subproblem of *DP*, the enhancement of this subproblem will be accumulated for SBX or DE in Lines 5-6. At the same time, the evolutionary potentiality (which will be introduced in Section III.D) of the updated subproblem will be reset to 1.0 in Line 8. Otherwise, if SI_i generated by DE can replace any solution of *PP*, the evolutionary potentiality of its parent's associated subproblem will also be reset to 1.0. At last, in Line 16, the updated F_{DE}, F_{SBX}, *DP*, and *PP* are returned.

D. Usage of the Population Update Information

Inspired by the utility function presented in MOEA/D-DRA [43], GRA [51] and EAG [36], the update information among the employed populations can be further used to select some potential subproblems of *DP*, hoping to ameliorate the subproblems of *DP* or to produce more promising offspring in *PP*. In this paper, the proposed evolutionary potentiality (EP) model for each subproblem in *DP* is defined as follows:

$$EP^i = \begin{cases} 1 & \textbf{if the solution linked to } \lambda^i \text{ is updated} \\ & \text{or the offspring from } \lambda^i \text{ can update } PP, \\ EP^i \times 0.95 & \textbf{otherwise} \end{cases} \quad (9)$$

where EP^i ($i = 1, 2, \ldots, N$) indicates the evolutionary potentiality of the i-th subproblem, and all the EP values are initialized to 1.0. As shown in the proposed EP model, three cases are considered: 1) if the i-th subproblem is updated successfully, its EP value will be assigned to a high value, hoping that this subproblem can be further enhanced; 2) if an offspring produced by the i-th subproblem can successfully update *PP*, its EP value is also assigned to a high value, encouraging the production of more promising offspring for *PP*; 3) otherwise, the EP value will be shrunk by a certain ratio at each generation. The advantages of this model are clarified as: 1) the population update information is used to guide the evolution in *DP*; and 2) it doesn't need to preset a threshold as required by the DRA strategy in [43].

IV. A Generalized Ensemble Framework

The proposed EF-PD can be further extended to be a more generalized EF. In detail, EF can use $J \geqslant 2$ populations, with J evolutionary operators and J selection criteria to run the competitive evolution and cooperative selection. As shown in Fig. 4, P_j, Op_j and SC_j indicate the j-th population, evolutionary operator and selection criterion, respectively, where j=1, 2, …, J. At the initialization stage, all J populations are randomly generated, and the computational resources (i.e., the number of executions of Op_j on P_j) are evenly assigned, subjected to the total number of execution (i.e., N) in one generation. After the evolutionary process, the credit assignment approach is used to evaluate the performance of each operator, such as the decomposition-based method using (5)-(7). Then, according to the operators' credits, the computational resources are reallocated again like (8) for the next generation. After that, all the new offspring solutions are shared and selected by all the employed populations. Before the next generation for competitive evolution and cooperative selection, the termination condition will be checked. If it is satisfied, one population specified by the user will be reported as the final result.

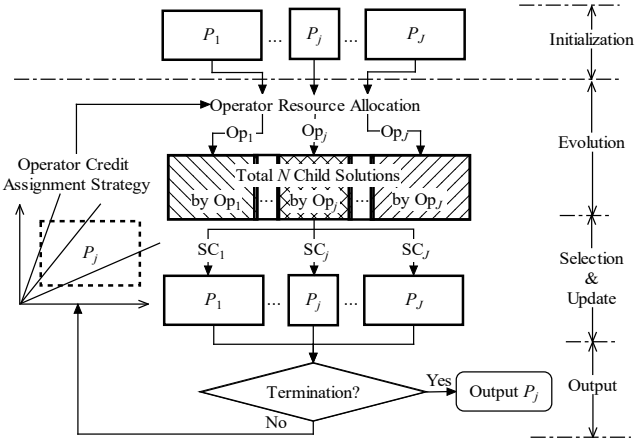


Fig. 4 Complete ensemble framework

A. Scalability Analysis

As shown in Fig. 4, the proposed EF has a strong ability for scalability. First of all, it can be easily scaled to use any number of populations to perform a competitive evolution and cooperative selection. Second, in order to keep the population's specific characteristics, each population will apply its own evolutionary operator and selection criterion, which may present some advantages on certain kinds of MOPs. Finally, the credit assignment approach in this framework can be replaced by any performance indicator that can fairly and effectively reflect the offspring's qualities, such as a Pareto-based method [16] or an indicator-based method [8]. It is worth noting that the decomposition-based credit assignment method is recommended in EF-PD due to its simplicity and efficiency.

B. Computational Complexity Analysis

Since any number of populations can be easily adopted in our framework, an important issue is the computational complexity of our approach. Here, our proposed EF-PD algorithm is taken as an example to assess the computational complexity at generation g.

- Mating selection

DP: the 10-tournament selection is run to select the potential subproblem and two other parents are randomly selected for DE. This process is all repeated N_{DE}^g times. Thus, the complexity in *DP* is approximately $O(10\, N_{DE}^g + 2\, N_{DE}^g) = O(12\, N_{DE}^g)$.

PP: two parents for SBX are randomly selected in the sorted *PP* during N_{SBX}^g times, therefore the complexity is $O(2\, N_{SBX}^g)$.

- Production

The complexities are $O(N_{DE}^g)$ and $O(N_{SBX}^g)$ for *DP* and *PP*, respectively, thus the total complexity is $O(N)$.

- Selection and update

DP: if the new solution is generated from *PP*, the search of the proper subproblem will take $O(N)$ basic operations and then updating its T neighboring solutions ($T<N$) will include $O(mT)$ basic operations. Also, the ideal point is updated by using $O(m)$ basic operations. Thus, the total number of operations is approximately $O(m\, N_{SBX}^g \times (N+mT+m) + m\, N_{DE}^g \times (mT+m))$, and the worst computational complexity is approximately $O(mN^2)$.

PP: As the total number of solutions is $2N$, the computational complexity for non-dominated sorting [6] is at most $O(4mN^2)$.

● Other operations

The total complexity of credit assignment and operator resource allocation is approximately $O(2N)$.

Therefore, the total computational complexity of EF-PD in generation g is approximately $O(12\ N_{DE}^{g} + 2\ N_{SBX}^{g} + N + mN^2 + 4mN^2 + 2N) \sim O(mN^2)$, while the computational complexities for *DP* and *PP* are also approximately $O(mN^2)$. Similarly, when more populations with computational complexity $O(mN^2)$ are to be used in the generalized EF, the total computational complexity will be kept at the same level with $O(mN^2)$. For the generalized cases in Fig. 4, the total computational complexity depends on the maximum computational complexity among populations with their different evolutionary operators and selection criteria.

V. Experimental Results

A. Test MOPs and Compared Algorithms

In this paper, thirty-one MOPs were tested, including the ZDT [52], WFG [42], DTLZ [53] and UF [43]-[44] test suites. The characteristics and parameter settings of these problems are shown in Table I, where m and d denote the numbers of objectives and decision variables, respectively. N and *max-ev* respectively indicate the population size and maximum number of function evaluations, which are adjusted according to the difficulties and complexities of different types of MOPs.

Five competitive algorithms are included for performance comparison. HEIA applies SBX and DE with a fixed ratio at each generation [26], while MOEA/D-DRA proposes a resource allocation strategy to optimize some potential subproblems [41]. FRRMAB [28] presents a credit-based selection method to choose the best operator among four DE variants. EAG [36] and BCE [38] employ two selection criteria for population update, namely the Pareto and decomposition-based selection criteria. As summarized in Table II, P_c is the crossover probability and P_m is the mutation probability; η_c and η_m are the distribution indexes of SBX and PM, respectively. For the DE operator, *CR* and *F* are the crossover rate and scaling factor, respectively, T denotes the size of the neighborhood for the weight vectors, δ is the probability to select the parents from T neighbors, and n_r is the maximum number of parent solutions to be updated by each child solution. Please note that the parameters in the adopted test MOPs and the compared algorithms are set as recommended in their original references.

B. Performance Indicators

Inverted generational distance (IGD) [48] and hypervolume (HV) [8], [54]-[55] are the performance indicators that we adopted to assess both convergence and diversity of the final solution set. When computing IGD or HV, only one population or archive is produced as the final solution set from the compared MOEAs with two populations (BCE, EAG and EF-PD), which ensures a fair comparison with respect to the approaches that only use one population (HEIA, MOEA/D-DRA, and FRRMAB). A small IGD value and a large HV value indicate better qualities of solution sets. Due to page limitations, the definitions of IGD and HV are provided in Section S-II of the supplementary file.

TABLE I
Characteristics and Parameter Settings of Test Problems [37]

MOPs	m	d	Characteristics	N	*max-ev*
ZDT1	2	30	Convex	100	2.5×10^4
ZDT2	2	30	Concave	100	2.5×10^4
ZDT3	2	30	Discontinuous	100	2.5×10^4
ZDT4	2	10	Convex, Multimodal	100	2.5×10^4
ZDT6	2	10	Concave, Multimodal, Biased	100	2.5×10^4
DTLZ1	3	7	Linear, Multimodal	500	1.0×10^5
DTLZ2	3	12	Concave, Sphere PF	500	1.0×10^5
DTLZ3	3	12	Concave, Sphere PF, Multimodal	500	1.0×10^5
DTLZ4	3	12	Concave, Biased	500	1.0×10^5
DTLZ5	3	12	Concave, Degenerate	500	1.0×10^5
DTLZ6	3	12	Concave, Degenerate, Biased	500	1.0×10^5
DTLZ7	3	22	Mixed, Discontinuous, Multimodal	500	1.0×10^5
WFG1	2	22	Mixed , Biased	200	1.0×10^5
WFG2	2	22	Convex, Discontinuous, Nonseparable	200	1.0×10^5
WFG3	2	22	Linear, Degenerate, Nonseparable	200	1.0×10^5
WFG4	2	22	Concave, Multimodal	200	1.0×10^5
WFG5	2	22	Concave, Deceptive	200	1.0×10^5
WFG6	2	22	Concave, Nonseparable	200	1.0×10^5
WFG7	2	22	Concave, Biased	200	1.0×10^5
WFG8	2	22	Concave, Nonseparable, Biased	200	1.0×10^5
WFG9	2	22	Concave, Nonsepar., Deceptive, Biased	200	1.0×10^5
UF1	2	30	Convex, Complex PS	300	3.0×10^5
UF2	2	30	Convex, Complex PS	300	3.0×10^5
UF3	2	30	Convex, Complex PS	300	3.0×10^5
UF4	2	30	Concave, Complex PS	300	3.0×10^5
UF5	2	30	Linear, Discrete, Complex PS	300	3.0×10^5
UF6	2	30	Linear, Discontinuous, Complex PS	300	3.0×10^5
UF7	2	30	Linear, Complex PS	300	3.0×10^5
UF8	3	30	Concave, Complex PS	600	3.0×10^5
UF9	3	30	Linear, Discontinuous, Complex PS	600	3.0×10^5
UF10	3	30	Concave, Complex PS	600	3.0×10^5

C. General Comparisons on All the Test MOPs

All the compared algorithms were run 30 independent times for each test MOP. The median value (i.e., the arithmetic mean of the 15th and 16th results in ascending order according to IGD) and the corresponding interquartile range (IQR) of IGD are listed in Table III, where the rank of each algorithm at each test MOP is included in the brackets and the average rank with respect to each series of test MOPs is provided. Moreover, in order to ensure statistical significance, the two-sided Wilcoxon's rank sum test [56]-[57] was performed between the results of each compared algorithm and EF-PD, by using the function "*ranksum*" in Matlab 2015, at a significance level of 5%.

1) Analysis on the ZDT Problems

As observed from Table III, EF-PD achieves the best results on ZDT1-ZDT4, while MOEA/D-DRA and FRRMAB obtain the 1st and 2nd best results on ZDT6. As the PF of ZDT6 is non-uniformly distributed [52], the diversity maintenance strategy in the Pareto-based selection criterion may have difficulties to cover a biased PF. The decomposition-based selection criterion in MOEA/D-DRA and FRRMAB performs much better, but they show poor performance on the rest of the ZDT problems, especially on ZDT3 with a discontinuous PF. Moreover, as HEIA maintains an operator hybridization strategy of mixing SBX and DE, it has a promising performance on all the ZDT problems. EF-PD further employs an adaptive ensemble approach, which governs the use of SBX and DE,

TABLE II
EVOLUTIONARY OPERATOR, SELECTION CRITERIA AND PARAMETERS SETTINGS OF ALL THE COMPARED ALGORITHMS

Algorithms	Evolutionary Operators	Selection Criteria	Parameter Settings
HEIA	SBX+PM, DE+PM	PC	$N=100, NA=20, P_c=1, P_m=1/n, \eta_c=\eta_m=20, CR=1, F=0.5, T=20, \delta=0.9$
DRA	DE+PM	DC	$N=100, CR=1, F=0.5, P_m=1/n, \eta_m=20, T=0.1N, \delta=0.9, n_r=0.01N$
FRRMAB	DE variants +PM	DC	$N=100, CR=1, F=0.5, P_m=1/n, \eta_m=T=20, \delta=0.9, n_r=2, C=5, W=0.5N, D=1$
EAG	SBX+PM	PC, DC	$N=100, CR=0.5, F=0.5, P_m=1/n, \eta_m=20, T=20$
BCE	Variation, Individual Exploration	PC, NPC	$N=100, CR=1, F=0.5, P_c=1, P_m=1/n, \eta_c=\eta_m=20, T=20,$
EF-PD	SBX+PM, DE+PM	PC, DC	$N=100, P_c=1, P_m=1/n, \eta_c=\eta_m=20, CR=1, F=0.5, T=20, \delta$=0.9, $n_r=2$

DRA, FRRMAB, EAG and BCE are short for MOEA/D-DRA, MOEA/D-FRRMAB, EAG-MOEA/D and BCE-MOEA/D+TCH, respectively. The DE variants include DE-rand-1/-2 and DE-current-to-rand-1/-2. PC and DC stand for the Pareto and decomposition-based selection criteria, respectively.

TABLE III
IGD RESULTS OF THE COMPARED ALGORITHMS ON ALL THE TEST PROBLEMS

MOPs	HEIA	MOEA/D-DRA	FRRMAB	EAG	BCE	EF-PD
ZDT1	$(3.9\pm0.084)\cdot10^{-3}(2^-)$	$(7.4\pm3.0)\cdot10^{-3}(5^-)$	$(3.7\pm1.0)\cdot10^{-2}(6^-)$	$(5.5\pm0.59)\cdot10^{-3}(4^-)$	$(4.3\pm0.12)\cdot10^{-3}(3^-)$	**$(3.8\pm0.056)\cdot10^{-3}(1)$**
ZDT2	$(4.0\pm0.07)\cdot10^{-3}(2^-)$	$(1.3\pm0.32)\cdot10^{-2}(5^-)$	$(3.1\pm0.91)\cdot10^{-2}(6^-)$	$(4.9\pm0.24)\cdot10^{-3}(4^-)$	$(4.2\pm0.24)\cdot10^{-3}(3^-)$	**$(3.8\pm0.053)\cdot10^{-3}(1)$**
ZDT3	$(4.4\pm0.068)\cdot10^{-3}(2^-)$	$(1.6\pm0.46)\cdot10^{-2}(5^-)$	$(7.7\pm3.7)\cdot10^{-2}(6^-)$	$(5.2\pm0.26)\cdot10^{-3}(4^-)$	$(4.7\pm0.074)\cdot10^{-3}(3^-)$	**$(4.4\pm0.054)\cdot10^{-3}(1)$**
ZDT4	$(3.9\pm0.19)\cdot10^{-3}(2\approx)$	$(4.9\pm11)\cdot10^{-2}(5^-)$	$(3.1\pm4.0)\cdot10^{-1}(6^-)$	$(5.3\pm0.87)\cdot10^{-3}(3^-)$	$(7.5\pm3.0)\cdot10^{-3}(4^-)$	**$(3.8\pm0.39)\cdot10^{-3}(1)$**
ZDT6	$(2.9\pm0.19)\cdot10^{-3}(3\approx)$	**$(2.4\pm0.029)\cdot10^{-3}(1^+)$**	$(2.4\pm0.034)\cdot10^{-3}(2^+)$	$(3.9\pm0.78)\cdot10^{-3}(5^-)$	$(1.2\pm0.2)\cdot10^{-2}(6^-)$	$(2.9\pm0.15)\cdot10^{-3}(4)$
B/S/W	0/2/3	1/0/4	1/0/4	0/0/5	0/0/5	——
Average Rank	2.2	4.2	5.2	4.0	3.8	**1.6**
WFG1	$(8.9\pm3.0)\cdot10^{-2}(2^-)$	$(9.3\pm0.67)\cdot10^{-1}(5^-)$	$(1.1\pm0.055)\cdot10^{0}(6^-)$	$(5.4\pm1.2)\cdot10^{-1}(3^-)$	$(6.0\pm0.52)\cdot10^{-1}(4^-)$	**$(4.9\pm1.7)\cdot10^{-2}(1)$**
WFG2	**$(5.4\pm0.49)\cdot10^{-3}(1^+)$**	$(2.4\pm0.13)\cdot10^{-2}(4^-)$	$(2.4\pm0.26)\cdot10^{-2}(3^-)$	$(5.2\pm5.6)\cdot10^{-2}(5^-)$	$(6.3\pm0.19)\cdot10^{-2}(6^-)$	$(5.5\pm0.55)\cdot10^{-3}(2)$
WFG3	$(8.2\pm0.71)\cdot10^{-3}(2^-)$	$(9.5\pm1.2)\cdot10^{-3}(5^-)$	$(8.6\pm0.35)\cdot10^{-3}(4^-)$	$(1.0\pm0.083)\cdot10^{-2}(6^-)$	$(8.2\pm1.2)\cdot10^{-3}(3^-)$	**$(6.4\pm0.67)\cdot10^{-3}(1)$**
WFG4	$(7.6\pm0.78)\cdot10^{-3}(3^-)$	$(4.5\pm1.1)\cdot10^{-2}(5^-)$	$(7.7\pm0.57)\cdot10^{-2}(6^-)$	$(2.5\pm0.45)\cdot10^{-2}(4^-)$	$(7.1\pm0.44)\cdot10^{-3}(2^-)$	**$(6.3\pm0.51)\cdot10^{-3}(1)$**
WFG5	$(6.6\pm0.028)\cdot10^{-2}(6^-)$	$(6.6\pm0.015)\cdot10^{-2}(4^-)$	$(6.6\pm0.0093)\cdot10^{-2}(3^-)$	$(6.6\pm0.0025)\cdot10^{-2}(5^-)$	$(6.6\pm0.0082)\cdot10^{-2}(2^-)$	**$(6.5\pm0.0057)\cdot10^{-2}(1)$**
WFG6	**$(3.4\pm2.7)\cdot10^{-2}(1\approx)$**	$(1.2\pm0.0039)\cdot10^{-1}(6^-)$	$(3.5\pm9.3)\cdot10^{-2}(2\approx)$	$(4.9\pm6.3)\cdot10^{-2}(4\approx)$	$(5.4\pm1.2)\cdot10^{-2}(5^-)$	$(4.4\pm2.0)\cdot10^{-2}(3)$
WFG7	$(7.4\pm0.34)\cdot10^{-3}(3^-)$	$(9.4\pm0.28)\cdot10^{-3}(6^-)$	$(9.2\pm0.16)\cdot10^{-3}(5^-)$	$(8.8\pm0.53)\cdot10^{-3}(4^-)$	$(6.9\pm0.28)\cdot10^{-3}(2^-)$	**$(6.1\pm0.083)\cdot10^{-3}(1)$**
WFG8	$(1.1\pm0.041)\cdot10^{-1}(3\approx)$	$(1.2\pm0.063)\cdot10^{-4}(4^-)$	**$(1.1\pm0.055)\cdot10^{-1}(1\approx)$**	$(1.3\pm0.067)\cdot10^{-1}(6^-)$	$(1.2\pm0.09)\cdot10^{-1}(5^-)$	$(1.1\pm0.059)\cdot10^{-1}(2)$
WFG9	$(1.2\pm0.0022)\cdot10^{-1}(6^-)$	$(1.2\pm0.0013)\cdot10^{-1}(4^-)$	$(1.2\pm0.99)\cdot10^{-1}(5^-)$	$(7.6\pm11)\cdot10^{-2}(2\approx)$	**$(1.4\pm0.99)\cdot10^{-2}(1^+)$**	$(1.2\pm0.00079)\cdot10^{-1}(3)$
B/S/W	1/2/6	0/0/9	0/2/7	0/2/7	1/0/8	——
Average Rank	3.0	4.8	3.9	4.3	3.3	**1.7**
DTLZ1	$(1.2\pm0.049)\cdot10^{-2}(4^-)$	$(1.2\pm0.06)\cdot10^{-2}(6^-)$	$(1.2\pm0.062)\cdot10^{-2}(5^-)$	$(1.1\pm0.035)\cdot10^{-2}(3^-)$	**$(8.9\pm0.62)\cdot10^{-3}(1^+)$**	$(1.1\pm0.02)\cdot10^{-2}(2)$
DTLZ2	$(3.1\pm0.085)\cdot10^{-2}(6^-)$	$(2.8\pm0.031)\cdot10^{-2}(2\approx)$	$(2.9\pm0.017)\cdot10^{-2}(5\approx)$	$(2.8\pm0.074)\cdot10^{-2}(4\approx)$	**$(2.3\pm0.032)\cdot10^{-2}(1^+)$**	$(2.8\pm0.074)\cdot10^{-2}(2)$
DTLZ3	$(3.0\pm0.073)\cdot10^{-2}(5^-)$	$(3.0\pm0.13)\cdot10^{-2}(4^-)$	$(3.1\pm4.7)\cdot10^{-2}(6^-)$	$(2.9\pm0.09)\cdot10^{-2}(3^-)$	**$(2.7\pm0.51)\cdot10^{-2}(1^+)$**	$(2.9\pm0.068)\cdot10^{-2}(2)$
DTLZ4	$(3.2\pm0.31)\cdot10^{-2}(6^-)$	$(2.2\pm0.24)\cdot10^{-2}(3^+)$	$(2.1\pm0.063)\cdot10^{-2}(2^+)$	$(3.1\pm0.37)\cdot10^{-2}(5\approx)$	**$(2.0\pm0.19)\cdot10^{-2}(1^+)$**	$(3.0\pm0.24)\cdot10^{-2}(4)$
DTLZ5	$(8.5\pm0.47)\cdot10^{-4}(2^-)$	$(2.9\pm0.22)\cdot10^{-3}(5^-)$	$(3.1\pm0.037)\cdot10^{-3}(6^-)$	$(1.1\pm0.05)\cdot10^{-3}(4^-)$	$(8.8\pm0.54)\cdot10^{-4}(3^-)$	**$(8.0\pm0.48)\cdot10^{-4}(1)$**
DTLZ6	$(8.1\pm0.6)\cdot10^{-4}(2\approx)$	$(2.8\pm0.09)\cdot10^{-3}(4^-)$	$(2.8\pm0.015)\cdot10^{-3}(5^-)$	$(1.0\pm0.082)\cdot10^{-3}(3^-)$	$(3.1\pm2.4)\cdot10^{-2}(6^-)$	**$(7.9\pm0.53)\cdot10^{-4}(1)$**
DTLZ7	$(3.2\pm0.18)\cdot10^{-2}(3\approx)$	$(8.9\pm0.19)\cdot10^{-2}(6^-)$	$(8.6\pm0.016)\cdot10^{-2}(5^-)$	$(3.3\pm0.19)\cdot10^{-2}(4\approx)$	**$(2.4\pm0.083)\cdot10^{-2}(1^+)$**	$(3.2\pm0.26)\cdot10^{-2}(2)$
B/S/W	0/2/5	1/1/5	1/1/5	0/3/4	5/0/2	——
Average Rank	4.0	4.3	4.9	3.7	**2.0**	**2.0**
UF1	$(2.6\pm0.17)\cdot10^{-3}(6^-)$	$(2.5\pm0.63)\cdot10^{-3}(4^-)$	$(1.8\pm0.18)\cdot10^{-3}(2^-)$	$(3.1\pm2.0)\cdot10^{-1}(5^-)$	$(2.3\pm0.12)\cdot10^{-3}(3^-)$	**$(1.7\pm0.098)\cdot10^{-3}(1)$**
UF2	$(5.5\pm0.54)\cdot10^{-3}(3\approx)$	$(8.0\pm4.6)\cdot10^{-3}(5\approx)$	**$(3.5\pm0.54)\cdot10^{-3}(1^+)$**	$(1.4\pm21.6)\cdot10^{-2}(6^-)$	$(7.2\pm1.3)\cdot10^{-3}(4^-)$	$(5.4\pm2.5)\cdot10^{-3}(2)$
UF3	$(1.5\pm0.89)\cdot10^{-2}(5^-)$	$(9.6\pm12)\cdot10^{-3}(4\approx)$	**$(2.1\pm1.4)\cdot10^{-3}(1^+)$**	$(2.7\pm2.1)\cdot10^{-2}(6^-)$	$(4.8\pm6.2)\cdot10^{-3}(2\approx)$	$(5.7\pm8.3)\cdot10^{-3}(3)$
UF4	$(3.8\pm0.059)\cdot10^{-2}(2^-)$	$(6.0\pm0.43)\cdot10^{-2}(4^-)$	$(5.8\pm0.72)\cdot10^{-2}(3^-)$	$(7.2\pm0.54)\cdot10^{-2}(5^-)$	$(9.4\pm0.87)\cdot10^{-2}(6^-)$	**$(3.7\pm0.052)\cdot10^{-2}(1)$**
UF5	$(2.0\pm1.1)\cdot10^{-1}(2\approx)$	$(3.1\pm2.0)\cdot10^{-1}(5^-)$	$(2.5\pm0.67)\cdot10^{-1}(3\approx)$	$(3.9\pm1.8)\cdot10^{-1}(6^-)$	$(2.6\pm0.97)\cdot10^{-1}(4^-)$	**$(1.8\pm1.0)\cdot10^{-1}(1)$**
UF6	$(1.8\pm0.58)\cdot10^{-1}(2\approx)$	$(1.9\pm0.97)\cdot10^{-1}(3\approx)$	**$(1.1\pm1.6)\cdot10^{-1}(1^+)$**	$(4.0\pm2.1)\cdot10^{-1}(6^-)$	$(1.9\pm0.62)\cdot10^{-1}(4\approx)$	$(2.0\pm2.2)\cdot10^{-1}(5)$
UF7	$(2.8\pm0.31)\cdot10^{-3}(5^-)$	$(2.3\pm0.45)\cdot10^{-3}(3\approx)$	**$(2.1\pm0.17)\cdot10^{-3}(1^+)$**	$(9.0\pm10)\cdot10^{-3}(6^-)$	$(2.7\pm0.41)\cdot10^{-3}(4^-)$	$(2.2\pm4.3)\cdot10^{-3}(2)$
UF8	$(1.9\pm0.25)\cdot10^{-1}(6^-)$	**$(5.2\pm2.5)\cdot10^{-2}(1^+)$**	$(5.6\pm0.95)\cdot10^{-2}(2^+)$	$(7.0\pm2.3)\cdot10^{-2}(4^+)$	$(6.7\pm1.3)\cdot10^{-2}(3^+)$	$(8.0\pm0.59)\cdot10^{-2}(5)$
UF9	$(4.6\pm1.3)\cdot10^{-1}(6^-)$	**$(3.9\pm11)\cdot10^{-2}(1^+)$**	$(3.9\pm11)\cdot10^{-2}(2^+)$	$(1.7\pm1.0)\cdot10^{-1}(5\approx)$	$(1.5\pm0.11)\cdot10^{-1}(3\approx)$	$(1.6\pm1.3)\cdot10^{-1}(4)$
UF10	$(6.5\pm1.6)\cdot10^{-1}(6^-)$	$(4.1\pm1.5)\cdot10^{-1}(2^-)$	$(4.8\pm0.91)\cdot10^{-1}(4^-)$	$(4.6\pm0.65)\cdot10^{-1}(3^-)$	$(5.2\pm0.95)\cdot10^{-1}(5^-)$	**$(2.6\pm0.42)\cdot10^{-1}(1)$**
B/S/W	0/3/7	2/4/4	6/1/3	1/1/8	1/3/6	——
Average Rank	4.3	3.2	**2.0**	5.2	3.8	2.5
Total Rank	3.5	4.1	3.7	4.4	3.3	**2.0**
Total *B/S/W*	1/9/21	4/5/22	8/4/19	1/6/24	7/3/21	——

The results are formatted as *Median*±*IQR* (*Rank* $^{+\text{ or }\approx\text{ or }-}$). "+, ≈, −" respectively denote the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to the Wilcoxon rank sum test at a 0.05 significance level. "*B/S/W*" stand for the accumulated numbers of "+, ≈, −", respectively. We show the best results in **boldface**.

leading to a superior performance over HEIA on these ZDT problems.

2) Analysis on the WFG Problems

As shown in Table III, EF-PD performs best on WFG1, WFG3-WFG5, and WFG7. For the rest of the WFG problems (WFG2, WFG6 and WFG9), EF-PD also shows an acceptable performance. It is observed that the single-population based MOEAs (MOEA/D-DRA and FRRMAB) perform worse than the bi-population based MOEAs with two evolutionary strategies (BCE, HEIA, and EF-PD). This indicates that a single evolutionary operator or a single selection criterion is not sufficient to properly solve these more difficult WFG problems.

To graphically assess the performance of EF-PD, the final solution sets corresponding to the 15th best IGD values of all the algorithms are plotted in Fig. 5 for WFG1. MOEA/D-DRA, FRRMAB, BCE, and EAG show a poor convergence, while EF-PD and HEIA perform much better by combining the search

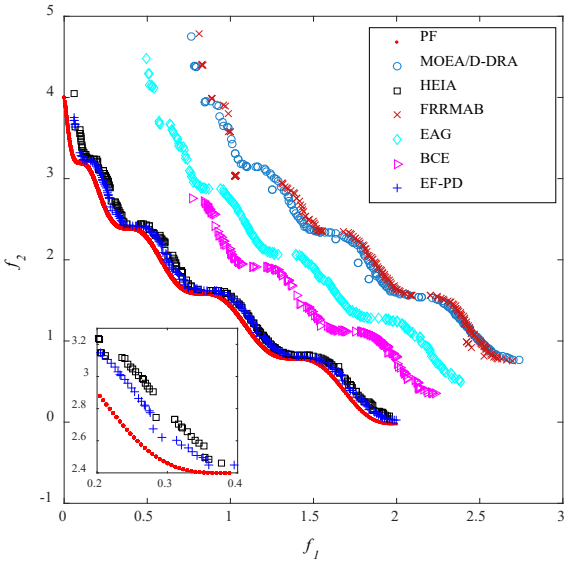


Fig. 5 Optimal approximate PFs found by all the algorithms on WFG1

patterns of SBX and DE, which confirms the effectiveness of hybridizing SBX and DE.

3) Analysis on the DTLZ Problems

From Table III, BCE achieves a better overall performance on all the DTLZ test problems, as the individual exploration strategy adopted in BCE is very effective to produce a good distribution of solutions in three-dimensional objective space. EF-PD performs a little worse than BCE but outperforms other algorithms. EF-PD obtains the best results on DTLZ5 and DTLZ6, and the second best results on DTLZ1-DTLZ3 and DTLZ7. This indicates that the algorithms with multiple selection criteria perform well in tackling the three-objective instances of the DTLZ problems. In detail, BCE and EF-PD can obtain both good convergence and diversity by using different selection criteria and by propagating the superior offspring among the populations. However, this propagation of solutions in EAG is unidirectional from the external archive to the evolved population, thus the cooperation between different populations in EAG is weaker than that of BCE and EF-PD.

When compared to EF-PD, BCE uses a niching strategy to measure the individuals' density, thus it performs very well on the three-objective test problems with generic spherical PFs [38], like DTLZ1-DTLZ4. However, EF-PD only uses the crowding-distance metric to reflect the individuals' density, which is more suitable for bi-objective problems [58]-[59]. When tackling DTLZ6 which has a degenerated PF, the optimal solution set found by BCE is expanded as a narrow belt covering the PF, whereas the results obtained by EF-PD better approximate the PFs, as shown in Fig. 6 with a local PF for DTLZ6. Moreover, the results on DTLZ5 (also with a degenerated PF) obtained by EF-PD and BCE are provided in Fig. S-6 of the supplementary file. Since DTLZ5 is easier to solve, EF-PD performs very well, while BCE still has some solutions diverging from the true PF. With respect to other competitors, which also use the crowding distance metric as their diversity maintenance strategy, the final solution set of EF-PD seems more promising on all the DTLZ problems.

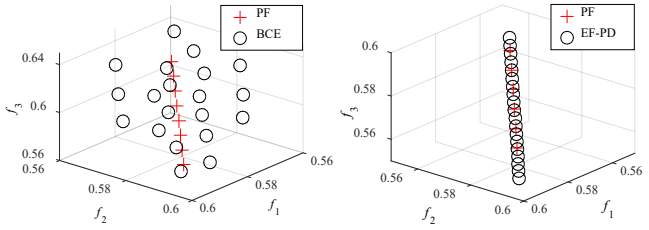


Fig. 6 The approximate local PFs found by BCE and EF-PD on DTLZ6

4) Analysis on the UF Test Problems

Since the UF test problems are characterized for having very complicated PS, the neighborhood structure employed by the decomposition-based approaches, such as MOEA/D-DRA and FRRMAB, is useful to strengthen the evolutionary search [7]. Thus, FRRMAB achieves the best average rank on all the UF problems, confirming that the DE operators (including four DE variants) can benefit from the neighborhood structure in a decomposition framework. Similarly, the performance of EF-PD also benefits from the decomposition-based population. As shown in Table III, EF-PD performs best on UF1, UF4-UF5 and UF10, and the total average rank shows that EF-PD also has a promising performance on all the UF test problems.

Based on all the IGD results reported in Table III, it is reasonable to conclude that EF-PD performs better than the compared algorithms in most cases, being able to tackle various kinds of complicated MOPs. Similar conclusions can be deduced from the HV comparison summary for all the test MOPs, as provided in Table S-I of the supplementary file due to page limitations.

D. Effectiveness of the Operator Competition Strategy

In order to study the effectiveness of the operator competition strategy, several EF-PD variants with fixed ratios of SBX and DE are realized to remove the operator competition in EF-PD. It should be noted that the cooperative selection of each EF-PD variant remained the same as that of EF-PD. For the sake of simplicity, these EF-PD variants are defined by the fixed ratio of DE, i.e., N_{DE}/N= 0%, 25%, 50%, 75%, and 100%, where N_{DE} is the number of executions of DE at one generation and N is the population size. All the other parameters are set the same as introduced in Section V.B. The comparison of results of EF-PD and its variants are collected in Table IV for all the test MOPs, while the summary of HV results is given in Table S-II of the supplementary file due to page limitations.

In detail, the EF-PD variant with 25% of DE achieves the 1st average rank on the ZDT problems, but performs worse than EF-PD on most of the UF problems. On the WFG problems, EF-PD obtains the best average rank, while the EF-PD variants with 25% and 50% of DE perform a little worse than EF-PD. Regarding the DTLZ problems, the EF-PD variant with 50% of DE and SBX outperforms the others, while a larger ratio of DE (such as 75% or 100% of DE) helps EF-PD to obtain more promising performance on the UF problems. According to these results, it is reasonable to conclude that a large number of execution times of SBX in *PP* is suitable for the ZDT test

TABLE IV
COMPARISON OF IGD RESULTS WITH TYPICAL FIXED RATIOS OF N_{DE} ON ALL THE TEST PROBLEMS

MOPs	N_{DE}/N=0%	N_{DE}/N=25%	N_{DE}/N=50%	N_{DE}/N=75%	N_{DE}/N=100%	EF-PD
ZDT1	**$(3.7\pm0.041)\cdot10^{-3}(1^+)$**	$(3.7\pm0.041)\cdot10^{-3}(2^+)$	$(3.7\pm0.06)\cdot10^{-3}(3^\approx)$	$(4.0\pm0.14)\cdot10^{-3}(5^-)$	$(1.5\pm0.42)\cdot10^{-2}(6^-)$	$(3.8\pm0.056)\cdot10^{-3}(4)$
ZDT2	$(3.8\pm0.045)\cdot10^{-3}(2^\approx)$	**$(3.8\pm0.053)\cdot10^{-3}(1^\approx)$**	$(3.8\pm0.05)\cdot10^{-3}(4^\approx)$	$(3.9\pm0.11)\cdot10^{-3}(5^-)$	$(1.2\pm0.75)\cdot10^{-2}(6^-)$	$(3.8\pm0.053)\cdot10^{-3}(3)$
ZDT3	$(4.4\pm0.049)\cdot10^{-3}(4^-)$	$(4.4\pm0.051)\cdot10^{-3}(2^\approx)$	$(4.4\pm0.035)\cdot10^{-3}(3^\approx)$	$(4.4\pm0.093)\cdot10^{-3}(5^-)$	$(5.3\pm2.9)\cdot10^{-2}(6^-)$	**$(4.4\pm0.054)\cdot10^{-3}(1)$**
ZDT4	$(4.1\pm1.1)\cdot10^{-3}(5^-)$	$(3.8\pm0.087)\cdot10^{-3}(2^+)$	**$(3.8\pm0.16)\cdot10^{-3}(1^+)$**	$(3.8\pm0.14)\cdot10^{-3}(4^\approx)$	$(1.3\pm1.6)\cdot10^{-1}(6^-)$	$(3.8\pm0.39)\cdot10^{-3}(3)$
ZDT6	$(3.0\pm0.087)\cdot10^{-3}(5^-)$	$(3.0\pm0.093)\cdot10^{-3}(3^-)$	$(3.0\pm0.12)\cdot10^{-3}(2^\approx)$	$(3.0\pm0.099)\cdot10^{-3}(4^-)$	$(3.0\pm0.19)\cdot10^{-3}(6^-)$	**$(2.9\pm0.15)\cdot10^{-3}(1)$**
Average Rank	3.7	**2.0**	2.6	4.6	6.0	2.4
WFG1	$(2.6\pm1.7)\cdot10^{-1}(4^-)$	$(6.8\pm1.8)\cdot10^{-2}(2^-)$	$(1.4\pm0.4)\cdot10^{-1}(3^-)$	$(2.9\pm0.5)\cdot10^{-1}(5^-)$	$(9.9\pm1)\cdot10^{-1}(6^-)$	**$(4.9\pm1.7)\cdot10^{-2}(1)$**
WFG2	$(4.6\pm5.5)\cdot10^{-2}(6^-)$	$(5.6\pm56)\cdot10^{-3}(3^\approx)$	$(5.5\pm56)\cdot10^{-3}(2^\approx)$	$(6.3\pm56)\cdot10^{-3}(4^-)$	$(1.3\pm0.48)\cdot10^{-2}(5^-)$	**$(5.5\pm0.55)\cdot10^{-3}(1)$**
WFG3	$(8.7\pm1.7)\cdot10^{-3}(5^-)$	**$(6.4\pm0.76)\cdot10^{-3}(1^\approx)$**	$(6.5\pm0.82)\cdot10^{-3}(3^-)$	$(6.6\pm0.55)\cdot10^{-3}(4^-)$	$(8.9\pm0.88)\cdot10^{-3}(6^-)$	$(6.4\pm0.67)\cdot10^{-3}(2)$
WFG4	$(6.4\pm0.51)\cdot10^{-3}(2^\approx)$	$(6.7\pm0.51)\cdot10^{-3}(3^-)$	$(7.9\pm0.93)\cdot10^{-3}(4^-)$	$(1.1\pm0.23)\cdot10^{-2}(5^-)$	$(4.8\pm0.43)\cdot10^{-2}(6^-)$	**$(6.3\pm0.51)\cdot10^{-3}(1)$**
WFG5	$(6.5\pm0.006)\cdot10^{-2}(4^\approx)$	**$(6.5\pm0.006)\cdot10^{-2}(1^\approx)$**	$(6.5\pm0.004)\cdot10^{-2}(3^\approx)$	$(6.5\pm0.005)\cdot10^{-2}(5^-)$	$(6.5\pm0.007)\cdot10^{-2}(6^-)$	$(6.5\pm0.005)\cdot10^{-2}(2)$
WFG6	$(5.7\pm1.3)\cdot10^{-2}(4^-)$	$(5.6\pm1.9)\cdot10^{-2}(3^\approx)$	$(4.5\pm1.5)\cdot10^{-2}(2^\approx)$	$(5.8\pm9.0)\cdot10^{-2}(5^-)$	$(1.2\pm0.0032)\cdot10^{-1}(6^-)$	**$(4.4\pm2.0)\cdot10^{-2}(1)$**
WFG7	$(6.0\pm0.13)\cdot10^{-3}(3^\approx)$	**$(6.0\pm0.13)\cdot10^{-3}(1^\approx)$**	$(6.1\pm0.1)\cdot10^{-3}(4^-)$	$(6.2\pm0.1)\cdot10^{-3}(5^-)$	$(7.5\pm0.45)\cdot10^{-3}(6^-)$	$(6.0\pm0.083)\cdot10^{-3}(2)$
WFG8	$(1.4\pm0.056)\cdot10^{-1}(6^-)$	$(1.1\pm0.025)\cdot10^{-1}(4^-)$	$(1.1\pm0.022)\cdot10^{-1}(2^\approx)$	**$(1.1\pm0.036)\cdot10^{-1}(1^\approx)$**	$(1.2\pm0.065)\cdot10^{-1}(5^-)$	$(1.1\pm0.059)\cdot10^{-1}(3)$
WFG9	$(1.2\pm0.0021)\cdot10^{-1}(6^-)$	$(1.2\pm0.0007)\cdot10^{-1}(5^\approx)$	**$(1.2\pm0.0008)\cdot10^{-1}(1^\approx)$**	$(1.2\pm0.001)\cdot10^{-1}(2^\approx)$	$(1.2\pm0.0007)\cdot10^{-1}(3^\approx)$	$(1.2\pm0.0007)\cdot10^{-1}(4)$
Average Rank	4.4	2.6	2.7	4	5.4	**1.9**
DTLZ1	$(1.1\pm0.064)\cdot10^{-2}(5^-)$	$(1.1\pm0.029)\cdot10^{-2}(3^\approx)$	$(1.1\pm0.03)\cdot10^{-2}(2^\approx)$	$(1.1\pm0.029)\cdot10^{-2}(4^-)$	$(1.3\pm1.4)\cdot10^{-2}(6^-)$	**$(1.1\pm0.02)\cdot10^{-2}(1)$**
DTLZ2	$(2.9\pm0.068)\cdot10^{-2}(6^-)$	$(2.8\pm0.068)\cdot10^{-2}(5^-)$	**$(2.8\pm0.07)\cdot10^{-2}(1^\approx)$**	$(2.8\pm0.035)\cdot10^{-2}(2^+)$	$(2.8\pm0.054)\cdot10^{-2}(3^+)$	$(2.8\pm0.074)\cdot10^{-2}(4)$
DTLZ3	$(3.0\pm0.096)\cdot10^{-2}(5^-)$	$(2.9\pm0.058)\cdot10^{-2}(3^\approx)$	$(2.9\pm0.05)\cdot10^{-2}(2^\approx)$	$(2.8\pm0.044)\cdot10^{-2}(4^-)$	$(5.7\pm13)\cdot10^{-2}(6^-)$	**$(2.9\pm0.068)\cdot10^{-2}(1)$**
DTLZ4	$(3.1\pm0.29)\cdot10^{-2}(3^\approx)$	$(3.1\pm0.36)\cdot10^{-2}(4^-)$	$(3.1\pm0.21)\cdot10^{-2}(2^\approx)$	$(3.1\pm0.21)\cdot10^{-2}(5^-)$	$(3.6\pm0.39)\cdot10^{-2}(6^-)$	**$(3.0\pm0.24)\cdot10^{-2}(1)$**
DTLZ5	$(8.1\pm0.46)\cdot10^{-4}(4^-)$	$(8.0\pm0.45)\cdot10^{-4}(2^\approx)$	$(8.0\pm0.55)\cdot10^{-4}(3^-)$	$(8.4\pm0.38)\cdot10^{-4}(5^-)$	$(9.1\pm0.4)\cdot10^{-4}(6^-)$	**$(8.0\pm0.48)\cdot10^{-4}(1)$**
DTLZ6	$(8.0\pm0.64)\cdot10^{-4}(6^-)$	$(8.0\pm0.36)\cdot10^{-4}(5^-)$	$(7.8\pm0.41)\cdot10^{-4}(2^\approx)$	**$(7.8\pm0.6)\cdot10^{-4}(1^\approx)$**	$(7.8\pm0.45)\cdot10^{-4}(3^\approx)$	$(7.9\pm0.53)\cdot10^{-4}(4)$
DTLZ7	$(3.2\pm0.12)\cdot10^{-2}(2^\approx)$	$(3.2\pm0.23)\cdot10^{-2}(4^\approx)$	**$(3.2\pm0.22)\cdot10^{-2}(1^\approx)$**	$(3.2\pm0.092)\cdot10^{-2}(5^\approx)$	$(3.3\pm0.19)\cdot10^{-2}(6^\approx)$	$(3.2\pm0.26)\cdot10^{-2}(3)$
Average Rank	4.4	3.7	**1.9**	3.7	5.4	2.1
UF1	$(7.4\pm2.4)\cdot10^{-2}(6^-)$	$(2.4\pm0.14)\cdot10^{-3}(5^-)$	$(2.0\pm0.097)\cdot10^{-3}(4^-)$	$(1.8\pm0.095)\cdot10^{-3}(3^\approx)$	$(1.8\pm0.13)\cdot10^{-3}(2^\approx)$	**$(1.7\pm0.098)\cdot10^{-3}(1)$**
UF2	$(2.7\pm0.66)\cdot10^{-2}(6^-)$	$(1.0\pm0.26)\cdot10^{-2}(5^-)$	$(7.8\pm3.3)\cdot10^{-3}(4^-)$	$(7.3\pm1.9)\cdot10^{-3}(3^-)$	$(5.4\pm2.3)\cdot10^{-3}(2^\approx)$	**$(5.4\pm2.5)\cdot10^{-3}(1)$**
UF3	$(1.6\pm0.59)\cdot10^{-1}(6^-)$	$(4.9\pm3.3)\cdot10^{-2}(5^-)$	$(1.7\pm2.6)\cdot10^{-2}(4^-)$	$(0.9\pm1.3)\cdot10^{-2}(3^-)$	$(7.2\pm8.8)\cdot10^{-3}(2^\approx)$	**$(5.7\pm8.3)\cdot10^{-3}(1)$**
UF4	$(4.2\pm0.058)\cdot10^{-2}(5^-)$	$(3.6\pm0.051)\cdot10^{-2}(2^+)$	**$(3.6\pm0.065)\cdot10^{-2}(1^+)$**	$(3.7\pm0.06)\cdot10^{-2}(4^-)$	$(6.1\pm0.65)\cdot10^{-2}(6^-)$	$(3.7\pm0.052)\cdot10^{-2}(3)$
UF5	$(2.6\pm1.7)\cdot10^{-1}(5^-)$	$(2.5\pm1.2)\cdot10^{-1}(4^-)$	$(2.2\pm2.4)\cdot10^{-1}(2^-)$	$(2.3\pm1.8)\cdot10^{-1}(3^-)$	$(4.4\pm2.8)\cdot10^{-1}(6^-)$	**$(1.8\pm1.0)\cdot10^{-1}(1)$**
UF6	$(3.0\pm2.1)\cdot10^{-1}(6^-)$	$(2.7\pm2.5)\cdot10^{-1}(5^-)$	$(2.5\pm1.6)\cdot10^{-1}(4^-)$	$(2.0\pm1.6)\cdot10^{-1}(2^\approx)$	$(2.2\pm3.9)\cdot10^{-1}(3^\approx)$	**$(2.0\pm2.2)\cdot10^{-1}(1)$**
UF7	$(7.6\pm28)\cdot10^{-2}(6^-)$	$(4.1\pm1.0)\cdot10^{-3}(5^-)$	$(3.1\pm2.4)\cdot10^{-3}(4^-)$	$(2.5\pm0.61)\cdot10^{-3}(3^\approx)$	$(2.3\pm0.91)\cdot10^{-3}(2^\approx)$	**$(2.2\pm4.3)\cdot10^{-3}(1)$**
UF8	$(8.5\pm8.0)\cdot10^{-2}(6^-)$	$(8.2\pm0.52)\cdot10^{-2}(5^-)$	$(8.0\pm0.59)\cdot10^{-2}(3^\approx)$	$(8.1\pm0.66)\cdot10^{-2}(4^\approx)$	**$(7.2\pm1.5)\cdot10^{-2}(1^\approx)$**	$(8.0\pm0.59)\cdot10^{-2}(2)$
UF9	$(1.9\pm1.0)\cdot10^{-1}(6^-)$	$(1.7\pm1.2)\cdot10^{-1}(5^\approx)$	$(7.2\pm13)\cdot10^{-2}(3^+)$	$(6.7\pm13)\cdot10^{-2}(2^+)$	**$(6.3\pm10)\cdot10^{-2}(1^+)$**	$(1.6\pm1.3)\cdot10^{-1}(4)$
UF10	$(2.7\pm1.1)\cdot10^{-1}(2^\approx)$	$(3.0\pm1.3)\cdot10^{-1}(4^-)$	$(3.0\pm1.3)\cdot10^{-1}(5^-)$	$(2.8\pm21)\cdot10^{-2}(3^+)$	$(5.0\pm1.0)\cdot10^{-1}(6^-)$	**$(2.6\pm0.42)\cdot10^{-1}(1)$**
Average Rank	5.4	4.5	3.4	3	3.1	**1.6**
Total Rank	4.6	3.4	2.7	3.7	4.7	**1.9**
B/*S*/*W*	1/6/24	3/13/15	3/16/12	3/9/19	2/9/20	/

The results are formatted as *Median*±*IQR* (*Rank* $^{+\text{ or }\approx\text{ or }-}$). "+,≈,−" respectively denote the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to the Wilcoxon rank sum test at a 0.05 significance level. "*B*/*S*/*W* " stand for the accumulated numbers of "+, ≈ , −", respectively. We show the best results in **boldface**.

problems, while a higher probability for executing DE in *DP* is recommended to solve the UF test problems.

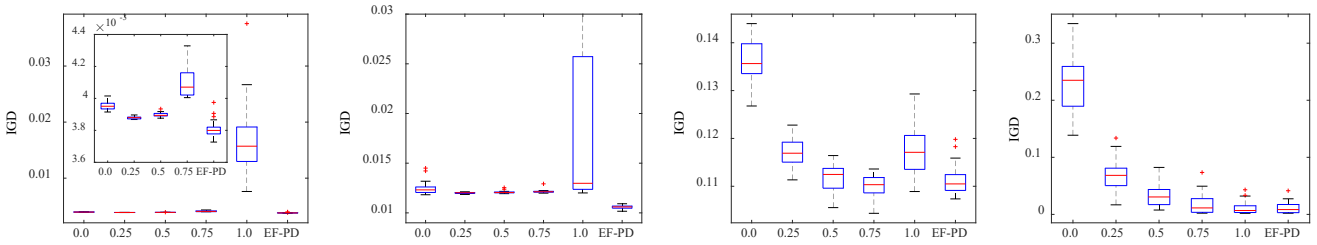
In Fig. 7, the detailed IGD results on some representative test MOPs (i.e., ZDT2, DTLZ1, WFG8 and UF3) are plotted. In this figure, the horizontal axis represents EF-PD and its typical variants with diverse ratios of DE, while the vertical axis shows the IGD values and their variations. It is found that different ratios of hybridizations of SBX and DE have some advantages on different MOPs, e.g., the variants with 20% and 50% of DE respectively perform best on ZDT2 and DTLZ1, while the variants with 75% and 100% of DE respectively give the best results on WFG8 and UF3. As a result, any EF-PD variant with a fixed ratio of operators can't always perform well on various types of MOPs. Thus, our proposed resource allocation strategy at different populations is necessary and effective.

The sequences of the average ratios N_{DE}/N (obtained from 30 runs of EF-PD) with respect to the generations are also plotted for all the test MOPs. Due to page limitations, they are shown in Section S-V of the supplementary file to illustrate the trade-off of SBX and DE during the execution. Particularly, the results typically obtained on ZDT2, DTLZ1, WFG8 and UF3 are plotted in Fig. 8, where N_{SBX} is the number of executions of SBX and the horizontal axis indicates the generations. The average ratios of N_{DE}/N on ZDT2 fluctuate below 0.5 in Fig. 8(a), which fits the results in Table IV that the best ratio of N_{DE}/N on ZDT2 may be some value in the range [0, 0.5]. On DTLZ1, the average ratio sequence of EF-PD has a large fluctuation and no evolutionary operator (SBX or DE) performs always well. As confirmed by Fig. 8(b), the adopted operator competition strategy is very effective and achieves the best performance. Moreover, taking WFG8 and UF3 respectively in Figs. 8(c) and 8(d) as examples, the optimal ratios of N_{DE}/N quickly climb and then stay at a high level of more than 0.8, which confirms that a high ratio of application of DE in *DP* with a small ratio of execution of SBX in *PP* are better to tackle these test MOPs. It should be noted that the tendency of ratios of N_{DE}/N on most of the UF test problems is very similar to that in Fig. 8(d), i.e., they all quickly go up and then gradually fall down, such as in UF1-UF3 and UF7-UF9 in Fig. S-5 of the supplementary file. This indicates that a large ratio of N_{DE}/N (i.e., more applications of DE) is required at the beginning for solving these test MOPs, and gradually SBX becomes important, as the ratio of N_{SBX}/N is enlarged.

TABLE V
COMPARISON OF IGD RESULTS AMONG EF-PD, EF-PD-N AND EF-PD-DRA

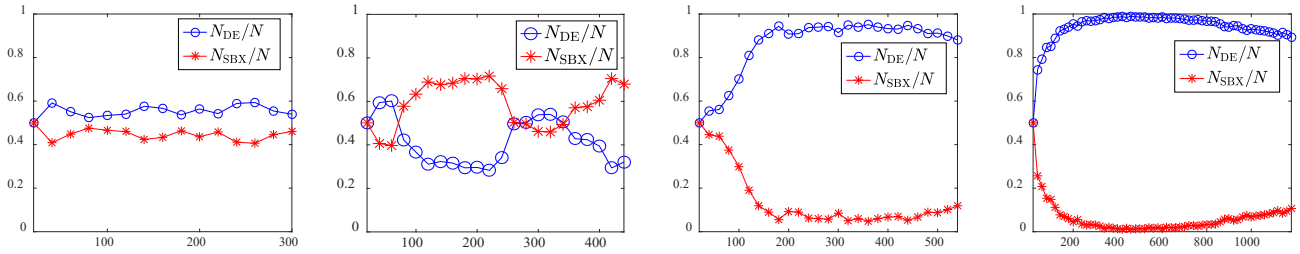
MOPs	EF-PD-N	EF-PD-DRA	EF-PD	MOPs	EF-PD-N	EF-PD-DRA	EF-PD
ZDT1	$(3.8\pm0.042)\cdot10^{-3}$(-)	$(3.79\pm0.056)\cdot10^{-3}$(-)	**$(3.75\pm0.056)\cdot10^{-3}$**	WFG5	$(6.53\pm0.0058)\cdot10^{-2}$(-)	$(6.53\pm0.0066)\cdot10^{-2}$(-)	**$(6.52\pm0.0057)\cdot10^{-2}$**
ZDT2	**$(3.83\pm0.06)\cdot10^{-3}$**(≈)	$(3.85\pm0.051)\cdot10^{-3}$(≈)	$(3.84\pm0.053)\cdot10^{-3}$	WFG6	$(4.5\pm1.9)\cdot10^{-2}$(≈)	$(4.58\pm2.8)\cdot10^{-2}$(≈)	**$(4.45\pm2.0)\cdot10^{-2}$**
ZDT3	$(4.38\pm0.072)\cdot10^{-3}$(-)	$(4.38\pm0.044)\cdot10^{-3}$(-)	**$(4.36\pm0.054)\cdot10^{-3}$**	WFG7	$(6.09\pm0.088)\cdot10^{-3}$(≈)	**$(6.04\pm0.16)\cdot10^{-3}$**(≈)	$(6.07\pm0.083)\cdot10^{-3}$
ZDT4	$(4.0\pm0.36)\cdot10^{-3}$(-)	$(3.94\pm0.31)\cdot10^{-3}$(≈)	**$(3.82\pm0.39)\cdot10^{-3}$**	WFG8	**$(1.11\pm0.033)\cdot10^{-1}$**(≈)	$(1.11\pm0.041)\cdot10^{-1}$(≈)	$(1.11\pm0.059)\cdot10^{-1}$
ZDT6	**$(2.92\pm0.17)\cdot10^{-3}$**(≈)	$(2.92\pm0.09)\cdot10^{-3}$(≈)	$(2.94\pm0.15)\cdot10^{-3}$	WFG9	$(1.24\pm0.0016)\cdot10^{-1}$(-)	$(1.24\pm0.0011)\cdot10^{-1}$(-)	**$(1.24\pm0.0008)\cdot10^{-1}$**
DTLZ1	$(1.06\pm0.021)\cdot10^{-2}$(≈)	$(1.06\pm0.015)\cdot10^{-2}$(≈)	**$(1.05\pm0.02)\cdot10^{-2}$**	UF1	$(1.8\pm0.091)\cdot10^{-3}$(-)	$(1.76\pm0.078)\cdot10^{-3}$(≈)	**$(1.72\pm0.099)\cdot10^{-3}$**
DTLZ2	$(2.88\pm0.061)\cdot10^{-2}$(-)	$(2.88\pm0.075)\cdot10^{-2}$(-)	**$(2.85\pm0.074)\cdot10^{-2}$**	UF2	$(5.82\pm1.7)\cdot10^{-3}$(≈)	$(5.52\pm1.2)\cdot10^{-3}$(≈)	**$(5.39\pm2.5)\cdot10^{-3}$**
DTLZ3	$(2.91\pm0.053)\cdot10^{-2}$(-)	$(2.91\pm0.042)\cdot10^{-2}$(-)	**$(2.86\pm0.068)\cdot10^{-2}$**	UF3	$(8.42\pm14)\cdot10^{-3}$(-)	$(6.86\pm9.5)\cdot10^{-3}$(-)	**$(5.73\pm8.3)\cdot10^{-3}$**
DTLZ4	**$(2.97\pm0.19)\cdot10^{-2}$**(≈)	$(3.04\pm0.21)\cdot10^{-2}$(≈)	$(2.99\pm0.24)\cdot10^{-2}$	UF4	$(3.66\pm0.067)\cdot10^{-2}$(≈)	**$(3.65\pm0.077)\cdot10^{-2}$**(≈)	$(3.65\pm0.052)\cdot10^{-2}$
DTLZ5	$(8.54\pm0.57)\cdot10^{-4}$(-)	$(8.36\pm0.45)\cdot10^{-4}$(≈)	**$(8.0\pm0.48)\cdot10^{-4}$**	UF5	$(2.47\pm2.5)\cdot10^{-1}$(-)	$(3.15\pm1.9)\cdot10^{-1}$(-)	**$(1.78\pm1.0)\cdot10^{-1}$**
DTLZ6	$(7.92\pm0.34)\cdot10^{-4}$(≈)	**$(7.83\pm0.63)\cdot10^{-4}$**(≈)	$(7.88\pm0.53)\cdot10^{-4}$	UF6	$(3.0\pm3.7)\cdot10^{-1}$(-)	$(2.14\pm2.4)\cdot10^{-1}$(≈)	**$(2.0\pm2.2)\cdot10^{-1}$**
DTLZ7	**$(3.11\pm0.15)\cdot10^{-2}$**(+)	$(3.13\pm0.17)\cdot10^{-2}$(≈)	$(3.19\pm0.26)\cdot10^{-2}$	UF7	$(2.17\pm0.29)\cdot10^{-3}$(≈)	**$(2.08\pm0.38)\cdot10^{-3}$**(≈)	$(2.18\pm4.3)\cdot10^{-3}$
WFG1	$(6.34\pm3.0)\cdot10^{-2}$(-)	$(6.3\pm1.8)\cdot10^{-2}$(-)	**$(4.88\pm1.7)\cdot10^{-2}$**	UF8	$(8.59\pm0.6)\cdot10^{-2}$(-)	$(8.49\pm0.85)\cdot10^{-2}$(≈)	**$(8.0\pm0.59)\cdot10^{-2}$**
WFG2	$(6.6\pm0.56)\cdot10^{-3}$(-)	$(5.93\pm0.56)\cdot10^{-3}$(-)	**$(5.53\pm0.55)\cdot10^{-3}$**	UF9	**$(5.54\pm10)\cdot10^{-2}$**(+)	$(1.5\pm0.9)\cdot10^{-1}$(≈)	$(1.55\pm1.3)\cdot10^{-1}$
WFG3	$(6.8\pm0.11)\cdot10^{-3}$(-)	$(6.78\pm0.84)\cdot10^{-3}$(-)	**$(6.4\pm0.67)\cdot10^{-3}$**	UF10	$(2.68\pm0.79)\cdot10^{-1}$(≈)	$(2.92\pm1.8)\cdot10^{-1}$(-)	**$(2.56\pm0.42)\cdot10^{-1}$**
WFG4	$(6.21\pm0.51)\cdot10^{-3}$(≈)	**$(6.17\pm0.42)\cdot10^{-3}$**(≈)	$(6.3\pm0.51)\cdot10^{-3}$	*B/S/W*	2/13/16	0/19/12	———

The results are formatted as *Median*±*IQR*(+ or ≈ or −). "+,≈,−" respectively denote the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to the Wilcoxon rank sum test at a 0.05 significance level. "*B*/*S*/*W*" stand for the accumulated numbers of "+, ≈ ,−", respectively. We show the best results in **boldface**.



(a) ZDT2 (b) DTLZ1 (c) WFG8 (d) UF3

Fig. 7 The box plots of IGD of EF-PD and its typical variants on some representative test problems



(a) ZDT2 (b) DTLZ1 (c) WFG8 (d) UF3

Fig. 8 The average ratio sequences of N_{DE}/N and N_{SBX}/N during the executions on some typical test problems

Overall, it is verified that no variant with fixed N_{DE}/N can be always good at tackling various MOPs. However, when taking an overview of all the used MOPs, EF-PD with the proposed resource allocation strategy is effective on adjusting the number of operator executions, to achieve a more promising performance in most cases.

E. Effectiveness of the Evolutionary Potentiality Model

In Section III.D, the population update information in EF-PD is embedded into the utility function used in MOEA/D-DRA [43], in order to enhance its scalability on multiple populations, as shown in the EP model. Comparisons between EF-PD and its two variants, namely EF-PD-N (EF-PD without any resource allocation) and EF-PD-DRA (EF-PD with the utility function used in MOEA/D-DRA [43]) are carried out to show the effectiveness of the EP model. Their IGD results are provided in Table V, which shows that EF-PD, EF-PD-N, and EF-PD-DRA obtain the best results on 20, 6, and 5 test problems, respectively. In detail, the Wilcoxon's rank sum test results reveal that EF-PD performs better than or similarly to EF-PD-N on 29 out of 31 test problems, which confirms the effectiveness of the proposed EP model. Also, EF-PD is significantly better than and statistically similar to EF-PD-DRA, respectively on 12 and 19 out of 31 test problems. Similar conclusions can be deduced from the HV results, which are collected in Table S-III of the supplementary file due to page limitations.

As we know, only the update information of a subproblem is used to model the utility function in MOEA/D-DRA [43]. However, due to the availability of multiple populations in EF-PD, a subproblem could generate superior solutions to update another population, and it should therefore be allocated more computational resources. Thus, this update information among different populations is recommended to be embedded into the DRA model (the utility function used in MOEA/D-DRA), forming our proposed EP model. Our experimental results show the superiority of our EP model over the DRA

TABLE VI
COMPARISON OF IGD RESULTS AMONG EF-PD-FIR, EF-PD-EFI AND EF-PD ON ALL THE TEST PROBLEMS

MOPs	EF-PD-FIR	EF-PD-EFI	EF-PD	MOPs	EF-PD-FIR	EF-PD-EFI	EF-PD
ZDT1	**(3.7±0.039)·10^{-3}**(+)	(3.8±0.079)·10^{-3}(≈)	(3.8±0.056)·10^{-3}	WFG5	(6.5±0.0038)·10^{-2}(≈)	(6.5±0.0052)·10^{-2}(≈)	**(6.5±0.0057)·10^{-2}**
ZDT2	**(3.8±0.035)·10^{-3}**(≈)	(3.8±0.079)·10^{-3}(≈)	(3.8±0.053)·10^{-3}	WFG6	(5.3±2.2)·10^{-2}(≈)	(4.6±2.7)·10^{-2}(≈)	**(4.4±2.0)·10^{-2}**
ZDT3	(4.4±0.036)·10^{-3}(≈)	(4.4±0.059)·10^{-3}(≈)	**(4.4±0.054)·10^{-3}**	WFG7	(6.1±0.16)·10^{-3}(≈)	**(6.0±0.09)·10^{-3}**(≈)	(6.1±0.083)·10^{-3}
ZDT4	(3.9±0.28)·10^{-3}(≈)	(3.9±0.27)·10^{-3}(≈)	**(3.8±0.39)·10^{-3}**	WFG8	**(1.1±0.032)·10^{-1}**(≈)	(1.1±0.072)·10^{-1}(≈)	(1.1±0.059)·10^{-1}
ZDT6	(3.0±0.06)·10^{-3}(≈)	**(2.9±0.12)·10^{-3}**(≈)	(2.9±0.15)·10^{-3}	WFG9	(1.2±0.0018)·10^{-1}(≈)	(1.2±0.0014)·10^{-1}(≈)	**(1.2±0.0008)·10^{-1}**
DTLZ1	(1.1±0.029)·10^{-2}(≈)	(1.1±0.017)·10^{-2}(-)	**(1.1±0.02)·10^{-2}**	UF1	(1.8±0.14)·10^{-3}(-)	**(1.7±0.082)·10^{-3}**(≈)	(1.7±0.098)·10^{-3}
DTLZ2	(2.8±0.051)·10^{-2}(≈)	(2.9±0.071)·10^{-2}(-)	**(2.8±0.074)·10^{-2}**	UF2	(7.8±2.0)·10^{-3}(-)	**(5.3±1.5)·10^{-3}**(≈)	(5.4±2.5)·10^{-3}
DTLZ3	**(2.9±0.061)·10^{-2}**(≈)	(2.9±0.11)·10^{-2}(≈)	(2.9±0.068)·10^{-2}	UF3	(1.3±2.5)·10^{-2}(-)	(7.9±11)·10^{-3}(≈)	**(5.7±8.3)·10^{-3}**
DTLZ4	**(2.9±0.21)·10^{-2}**(≈)	(3.0±0.18)·10^{-2}(≈)	(3.0±0.24)·10^{-2}	UF4	**(3.6±0.054)·10^{-2}**(+)	(3.7±0.067)·10^{-2}(-)	(3.7±0.052)·10^{-2}
DTLZ5	(8.1±0.59)·10^{-4}(≈)	(8.6±0.47)·10^{-4}(-)	**(8.0±0.48)·10^{-4}**	UF5	(2.4±1.1)·10^{-1}(-)	(2.4±2.0)·10^{-1}(-)	**(1.8±1.0)·10^{-1}**
DTLZ6	(8.0±0.57)·10^{-4}(≈)	**(7.8±0.64)·10^{-4}**(≈)	(7.9±0.53)·10^{-4}	UF6	(3.0±4.0)·10^{-1}(≈)	(2.6±1.7)·10^{-1}(≈)	**(2.0±2.2)·10^{-1}**
DTLZ7	**(3.2±0.25)·10^{-2}**(≈)	(3.2±0.18)·10^{-2}(≈)	(3.2±0.26)·10^{-2}	UF7	(2.6±0.91)·10^{-3}(-)	(2.2±0.52)·10^{-3}(≈)	**(2.2±4.3)·10^{-3}**
WFG1	(5.9±1.7)·10^{-2}(-)	(5.1±1.9)·10^{-2}(≈)	**(4.9±1.7)·10^{-2}**	UF8	(8.0±1.1)·10^{-2}(≈)	(8.2±1.2)·10^{-2}(≈)	**(8.0±0.59)·10^{-2}**
WFG2	(3.4±5.6)·10^{-2}(-)	(5.9±0.95)·10^{-3}(≈)	**(5.5±0.55)·10^{-3}**	UF9	**(7.5±12)·10^{-2}**(≈)	(1.6±1.1)·10^{-1}(≈)	(1.6±1.3)·10^{-1}
WFG3	(6.9±0.87)·10^{-3}(≈)	**(6.3±0.93)·10^{-3}**(≈)	(6.4±0.67)·10^{-3}	UF10	(2.8±1.5)·10^{-1}(≈)	(2.9±0.99)·10^{-1}(-)	**(2.6±0.42)·10^{-1}**
WFG4	(6.3±0.46)·10^{-3}(≈)	**(6.2±0.7)·10^{-3}**(≈)	(6.3±0.51)·10^{-3}	*B/S/W*	2/22/7	0/25/6	

The results are formatted as *Median*±*IQR*(+ or ≈ or −). "+,≈,−" respectively denote the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to the Wilcoxon rank sum test at a 0.05 significance level. "*B*/*S*/*W* " stand for the accumulated numbers of "+, ≈ ,− ", respectively. We show the best results in **boldface**.

TABLE VII
FINAL IGD COMPARISONS OF EF-PD-R, EF-PD-H, EF-PD-E WITH EF-PD

Problems \ Algorithms		EF-PD-R	EF-PD-H	EF-PD-E
ZDTs	*B/S/W*	1/4/0	2/2/1	0/1/4
WFGs	*B/S/W*	2/4/3	2/4/3	3/1/5
DTLZs	*B/S/W*	0/5/2	1/5/1	1/2/4
UFs	*B/S/W*	0/5/5	0/3/7	0/1/9
Total	*B/S/W*	3/18/10	5/14/12	4/5/22

B/*S*/*W* respectively denote the number of MOPs on that the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to Wilcoxon rank sum test at a 0.05 significance level.

model, and also confirm the usefulness of this update information among different populations.

VI. FURTHER DISCUSSIONS

A. Credit Assignment Strategies

One open issue in this work is the credit assignment approach [31]. According to the inputs, there are several kinds of credit assignment approaches [32], such as offspring versus parent (OP), set improvement (SI), and contribution to the set (CS). The extreme dynamic multi-armed bandit (Ex-DMAB) [19] belongs to the type of OP, which assigns the credit using the extreme value of the fitness improvement (EFI) of one operator. In the most relevant approaches, OP, SI and CS are hybridized to reward the operators, e.g., FRRMAB [28] uses the fitness improvement rate (FIR) combined by OP and SI, while MOEA/D-HH [60] employs three measure functions including OP and CS. In contrast, the average fitness improvement (AFI) employed by EF-PD only contains SI, thus it is very simple and efficient. Here, two other well-known credit assignment strategies (FIR [28] and EFI [19]) were also studied under the proposed EF. FIR and EFI are respectively embedded into EF-PD to substitute AFI, and their variants are respectively named EF-PD-FRI and EF-PD-EFI. To clarify their behavior, FIR and EFI are respectively defined by (10) and (11), as follows:

$$\Delta f^{i,FIR}(\mathbf{y}\,|\,\mathbf{x},\lambda^i,\mathbf{z}^*)=\frac{g^{tch}(\mathbf{x}\,|\,\lambda^i,\mathbf{z}^*)-g^{tch}(\mathbf{y}\,|\,\lambda^i,\mathbf{z}^*)}{g^{tch}(\mathbf{x}\,|\,\lambda^i,\mathbf{z}^*)}, \quad (10)$$

$$F^{EFI}_{\mathrm{Op}_j}=\max_{\mathbf{y}\in SI^{\mathrm{Op}_j}}\{g^{tch}(\mathbf{x}\,|\,\lambda^i,\mathbf{z}^*)-g^{tch}(\mathbf{y}\,|\,\lambda^i,\mathbf{z}^*)\}, \quad (11)$$

where **x** and **y** stand for the original and new associated solutions, respectively. $\Delta f^{i,FIR}(\mathbf{y}\,|\,\mathbf{x},\lambda^i,\mathbf{z}^*)$ denotes the improvement rate achieved by the new solution **y** over **x**, and then an average value will be calculated by (5) with respect to each operator as its credit. In (11), $SI^{\mathrm{Op}j}$ is the set of new associated solutions generated by the operator Op_j, and $F^{EFI}_{\mathrm{Op}_j}$ denotes the extreme (maximum) fitness improvement achieved by the operator Op_j, as its credit.

All the IGD results are collected in Table VI. EF-PD-FIR and EF-PD-EFI perform very similarly to EF-PD, as they have 22 and 25 statistically similar results out of 31 test problems. This is reasonable as they are very similar to show the operator's performance. Especially, the only difference between equations (10) in EF-PD-FIR and (6) in EF-PD is the additional denominator in equation (10). It leads to the slight difference on the IGD results as the denominators are often different for all the subproblems. According to the overall performance comparison, EF-PD shows slight advantages over EF-PD-FIR and EF-PD-EFI. The above experiments confirm that EF is scalable to embed various credit assignment approaches. That is to say, any performance indicator can be used in EF, if it is effective to reflect the operator's performance.

B. Matching Strategies of Operators and Populations

In EF, each evolutionary operator has a one-to-one matching with the population, i.e., SBX is only executed on *PP*, while DE is only run on *DP*. As introduced in Section IV.A, each population in Fig. 4 uses its exclusive evolutionary operator and selection criterion in order to present the advantages on certain kinds of MOPs. To study the rationality of the one-to-one matching strategy, EF-PD is further compared to its two variants with a random matching strategy (termed EF-PD-R) and a uniform matching strategy (termed EF-PD-H). In EF-PD-R, *PP* and *DP* will randomly select their evolutionary operators (SBX or DE) with a probability of 0.5 for all the individuals at each generation; whereas, in EF-PD-H, *PP* and *DP* will randomly select the evolutionary operator (SBX or DE) for each of its

TABLE VIII
COMPARISON OF IGD RESULTS BETWEEN EF-PDI AND EF-PD ON ALL THE TEST PROBLEMS

MOPs	EF-PDI	EF-PD	MOPs	EF-PDI	EF-PD	MOPs	EF-PDI	EF-PD
ZDT1	**$(3.7\pm0.048)\cdot10^{-3}$**(+)	$(3.8\pm0.056)\cdot10^{-3}$	WFG1	**$(4.2\pm0.21)\cdot10^{-2}$**(+)	$(4.9\pm1.7)\cdot10^{-2}$	UF1	**$(1.6\pm0.056)\cdot10^{-3}$**(+)	$(1.7\pm0.098)\cdot10^{-3}$
ZDT2	**$(3.8\pm0.058)\cdot10^{-3}$**(+)	$(3.8\pm0.053)\cdot10^{-3}$	WFG2	$(5.5\pm25)\cdot10^{-3}$(≈)	$(5.5\pm0.55)\cdot10^{-3}$	UF2	$(6.6\pm1.4)\cdot10^{-3}$(-)	**$(5.4\pm2.5)\cdot10^{-3}$**
ZDT3	$(4.3\pm0.088)\cdot10^{-3}$(≈)	$(4.4\pm0.054)\cdot10^{-3}$	WFG3	**$(5.3\pm0.8)\cdot10^{-3}$**(+)	$(6.4\pm0.67)\cdot10^{-3}$	UF3	$(6.3\pm12)\cdot10^{-3}$(≈)	$(5.7\pm8.3)\cdot10^{-3}$
ZDT4	$(3.8\pm0.15)\cdot10^{-3}$(≈)	$(3.8\pm0.39)\cdot10^{-3}$	WFG4	**$(5.4\pm0.14)\cdot10^{-3}$**(+)	$(6.3\pm0.51)\cdot10^{-3}$	UF4	**$(3.5\pm0.074)\cdot10^{-2}$**(+)	$(3.7\pm0.052)\cdot10^{-2}$
ZDT6	$(3.0\pm0.17)\cdot10^{-3}$(≈)	$(2.9\pm0.15)\cdot10^{-3}$	WFG5	**$(6.5\pm0.004)\cdot10^{-2}$**(+)	$(6.5\pm0.0057)\cdot10^{-2}$	UF5	$(2.1\pm2.1)\cdot10^{-1}$(-)	**$(1.8\pm1.0)\cdot10^{-1}$**
DTLZ1	$(1.1\pm0.021)\cdot10^{-2}$(≈)	$(1.1\pm0.02)\cdot10^{-2}$	WFG6	$(4.9\pm1.4)\cdot10^{-2}$(≈)	$(4.4\pm2.0)\cdot10^{-2}$	UF6	$(2.6\pm1.7)\cdot10^{-1}$(≈)	$(2.0\pm2.2)\cdot10^{-1}$
DTLZ2	$(2.8\pm0.053)\cdot10^{-2}$(≈)	$(2.8\pm0.074)\cdot10^{-2}$	WFG7	**$(5.9\pm0.17)\cdot10^{-3}$**(+)	$(6.1\pm0.083)\cdot10^{-3}$	UF7	$(2.5\pm0.61)\cdot10^{-3}$(≈)	$(2.2\pm4.3)\cdot10^{-3}$
DTLZ3	$(2.9\pm0.068)\cdot10^{-2}$(≈)	$(2.9\pm0.068)\cdot10^{-2}$	WFG8	**$(1.1\pm0.03)\cdot10^{-1}$**(+)	$(1.1\pm0.059)\cdot10^{-1}$	UF8	$(8.4\pm0.94)\cdot10^{-2}$(≈)	$(8.0\pm0.59)\cdot10^{-2}$
DTLZ4	$(3.0\pm0.19)\cdot10^{-2}$(≈)	$(3.0\pm0.24)\cdot10^{-2}$	WFG9	**$(1.2\pm0.0003)\cdot10^{-1}$**(+)	$(1.2\pm0.0008)\cdot10^{-1}$	UF9	**$(1.1\pm1.2)\cdot10^{-1}$**(+)	$(1.6\pm1.3)\cdot10^{-1}$
DTLZ5	$(8.2\pm0.57)\cdot10^{-4}$(≈)	$(8.0\pm0.48)\cdot10^{-4}$	DTLZ7	$(3.2\pm0.23)\cdot10^{-2}$(≈)	$(3.2\pm0.26)\cdot10^{-2}$	UF10	$(3.2\pm1.5)\cdot10^{-1}$(≈)	$(2.6\pm0.42)\cdot10^{-1}$
DTLZ6	$(7.7\pm0.45)\cdot10^{-4}$(≈)	$(7.9\pm0.53)\cdot10^{-4}$	*Total B/S/W*:	12/17/2				

The results are formatted as *Median*±*IQR*(+ or ≈ or −). "+,≈,−" respectively denote the performance of the compared algorithm is better than, similar with or worse than that of EF-PD, according to the Wilcoxon rank sum test at a 0.05 significance level. "*B*/*S*/*W* " stand for the accumulated numbers of "+, ≈ , −", respectively. We show the best results in **boldface**.

individuals according to a probability of 0.5. Moreover, an exchanged matching strategy is performed by applying SBX only on *DP* and DE only on *PP*, termed EF-PD-E. In the three variants of EF-PD, the mating selection strategies are not coupled with the evolutionary operators (SBX or DE), but associated to the populations (*PP* or *DP*), aiming to allow a fair comparison. Due to page limitations, the details of the mating selection strategies adopted in EF-PD-R, EF-PD-H and EF-PD-E are clarified in Table S-IV of the supplementary file.

The summarized IGD results are listed in Table VII, while the detailed IGD comparisons are provided in Table S-V of the supplementary file. It is confirmed that EF-PD with a one-to-one matching strategy of operators and populations performs better than EF-PD-R, EF-PD-H and EF-PD-E. Since DE can benefit from the neighborhood structure of subproblems under a decomposition-based framework, it is actually not so effective to execute DE on *PP*. Thus, the three EF-PD variants with different matching strategies of operators and populations are not recommended. EF-PD-E has the worst performance, validating the above statement and showing the rationale for employing SBX and DE respectively on *PP* and *DP* in EF-PD. In summary, to obtain a more promising performance under EF, it is suggested that each population should have its specific characteristic by choosing its preferred evolutionary operator and selection criterion.

C. Another EF Paradigm with Triple Populations

In order to show the high scalability of EF, an extended paradigm with three independent populations was designed, termed EF-PDI, where "I" indicates an **I**ndicator-based population embedded in EF. This additional population employs an HV-indicator-based selection criterion and SBX as in IBEA [8]. Table VIII shows the IGD results obtained by EF-PDI and EF-PD. As shown, EF-PDI performs better than or similarly to EF-PD on 29 out of 31 test problems, which indicates that the additional indicator-based population is able to further enhance EF-PD. Moreover, the adopted resource allocation strategy is effective to assign the computational resources to the population that best fits to solve the target MOP.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, an effective ensemble algorithm, EF-PD, and its generalized framework EF were proposed to perform a competition on various evolutionary operators and a cooperation of different selection criteria. In the proposed EF, a decomposition-based credit assignment approach and an operator resource allocation strategy are employed to drive the competitive running of multiple evolutionary operators, while the migration of superior offspring is carried out by using a cooperative selection mechanism. Thirty-one test MOPs were used to evaluate the performance of EF-PD, and the experimental results indicate the superiority of EF-PD over other compared algorithms in most cases. Also, the advantages of the operator resource allocation strategy and the proposed EP model were experimentally confirmed. Moreover, further discussions on EF were brought forward to study the rationale of the proposed EF-PD, and a new EF instance extended with triple populations was introduced to show its scalability.

As part of our future work, we are interested in embedding more evolutionary operators, selection criteria, and other resource allocation strategies into the proposed EF. Moreover, the parallel implementation of EF in hardware is another research path that we would like to explore in the future.

REFERENCES

[1] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, USA, 2001.

[2] C. A. Coello Coello, G. B. Lamont and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edtion, Springer, New York, 2007.

[3] K. C. Tan, S. C. Chiam, A. A. Mamun, and C. K. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 701-713, 2009.

[4] J. Branke, K. Deb, K. Miettinen and R. Słowiński, *Multiobjective Optimization. Lecture Notes in Computer Science*, vol 5252, Springer, Berlin, 2008.

[5] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *Swiss Federal Institute of Technology (ETH)*, Tech. rep., 103, Zurich, Switzerland, 2001.

[6] K. Deb, A. Pratap, S. Agarwal and R. Słowiński, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.

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

[8] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in book: *Parallel Problem Solving from Nature-PPSN VIII*, Springer Berlin Heidelberg, Sep. 2004, pp.832-842.

[9] J. A. Soria-Alcaraz, G. Ochoa, M. A. Sotelo-Figeroa, and E. K. Burke, "A methodology for determining an effective subset of heuristics in selection hyper-heuristics", *Eur. J. Oper. Res.*, 260(3), pp. 972-983, 2017.

[10] M. F. Tasgetiren, P. N. Suganthan and Q. K. Pan, "An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem", *Appl. Soft Comput.*, vol. 215, no. 9, pp. 3356-3368, 2010.

[11] S. Z. Zhao, P. N. Suganthan and Q. Zhang, "Decomposition based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes", *IEEE Trans. on Evol. Comput.*, vol. 16, no. 3, pp. 442-446, 2012.

[12] E. L. Yu and P. N. Suganthan, "Ensemble of niching algorithms", *Information Sciences*, vol. 180, no. 15, pp. 2815-2833, 2010.

[13] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques", *IEEE Trans. on Evol. Comput.*, vol. 14, no. 4, pp. 561-579, 2010.

[14] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms," *Caltech Concurrent Computation Program, California Instit.*, Tech. rep. 826, CA, USA, 1989.

[15] C. K. Goh, K. C. Tan, D. S. Liu and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 42-54, 2010.

[16] J. A. Vrugt, B. A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search", in *Proc. of* the *National Academy of Sciences*, 2007, vol. 104, no.3, pp. 708-711.

[17] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems,* vol. 9, no. 2, pp. 115-48, 1995.

[18] W. K. Mashwani and A. Salhi. "Multiobjective evolutionary algorithm based on multimethod with dynamic resources allocation." *Applied Soft Computing*, vol. 39(C), pp.292-309. 2016.

[19] A. Fialho, L. Da Costa, M. Schoenauer and M. Sebag, "Dynamic multiarmed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms," in *Proc. 3rd Int. Conf. Learning Intell. Optimization*, LNCS 5851. 2009, pp. 176–190.

[20] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Boston, USA: Kluwer, 2001.

[21] V. A. Shim, K. C. Tan and H. J. Tang, "Adaptive memetic computing for evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 610-621, 2015.

[22] Z. Zhan, Y. Li and J. Zhang, "Cloudde: A heterogeneous differential evolution algorithm and its cloud version", *IEEE Trans on Pa. & Dis.*, vol. 28(3), pp.704-716, 2017.

[23] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences,* vol. 329, pp. 329-345, 2016.

[24] Q. Zhu, Q. Lin, Z. Du, Z. Liang, W. Wang, Z. Zhu, J. Chen, P. Huang and Z. Ming, "A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm," *Information Sciences*, vol. 345, pp. 177-198, 2016.

[25] Y. Qi, Z. Hou, M. Yin, H. Sun and J. Huang, "An immune multi-objective optimization algorithm with differential evolution inspired recombination," *Appl Soft Comput.*, vol. 29, pp. 395-410, 2015.

[26] Q. Lin, J. Chen, Z. Zhan, W. Chen, C. A. Coello Coello, Y. Yin, C. Lin and J. Zhang, "A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems," *IEEE Trans. Evol. Comput.,* vol. 20, no. 5, pp. 711-729, 2016.

[27] J. H. Drake, E. Ozcan, E. K. Burke, "An improved choice function heuristic selection for cross domain heuristic search", In: *PPSN 2012, Part II. LNCS*, vol. 7492, pp. 307-316, 2012.

[28] K. Li, Á. Fialho, S. Kwong and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, v.18, no. 1, pp. 114-130, 2014.

[29] Q. Lin, Z. Liu, Q. Yan, Z. Du, C. A. Coello Coello, Z. Liang, W. Wang and J. Chen, "Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm," *Information Sciences*, vol. 339(C), pp. 332-352, 2016.

[30] C. K. Goh, Y. S. Ong and K. C. Tan. *Multi-Objective Memetic Algorithms-Studies in Computational Intelligence*, vol. 171, Springer, Verlag Berlin Heidelberg, 2009.

[31] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proc. ICGA*, 1989, pp. 61–69.

[32] N. Hitomi and D. Selva, "A classification and comparison of credit assignment strategies in multiobjective adaptive operator selection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 294-314, 2017.

[33] K. C. Tan, T. H. Lee, Y. J. Yang and D. S. Liu, "A cooperative coevolutionary algorithm for multiobjective optimization," *IEEE International Conf. on Syst., Man and Cybern.*, vol.2, pp. 1926-1931, 2004.

[34] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimization algorithm: The two-archive algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, vol. 1. Guangzhou, China, 2006, pp. 286-291.

[35] H. Wang, L. Jiao and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.,* vol. 19, no. 4, pp. 524-541, 2015.

[36] X. Cai, Y. Li, Z. Fan and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.,* vol. 19, no. 4, pp. 508-523, 2015.

[37] N. Al-Moubayed, A. Petrovski and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47-77, 2014.

[38] 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.

[39] K. Deb and H. G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *IEEE Trans. Evol. Comput*., 9(2), pp. 197-221, 2001.

[40] J. J. Durillo and A. J. Nebro, "jMetal: a Java framework for multi-objective optimization," *Adv. Eng. Softw*., vol. 42, no. 10, pp. 760-771, 2011.

[41] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput*., v. 15, no. 1, pp. 4-31, 2011.

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

[43] Q. Zhang, W. Liu and H. Li, "The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," in *IEEE Congr. on Evol. Comput.*, 2009, pp. 203-208.

[44] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *Technical Report CES-487*, University of Essex, 2009.

[45] L. Ke, Q. Zhang and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808-1820, 2014.

[46] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.,* vol. 18, no. 19, pp. 577-601, 2014.

[47] K. Li, K. Deb, Q. Zhang and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, 19(5), pp. 694-716, 2015.

[48] Y. Yuan, H. Xu, and B. Wang, "An experimental investigation of variation operators in reference-point based many-objective optimization", *2015 Annual Conf. on Genetic and Evolut. Comput.*, pp. 775-782.

[49] R. A. Gonçalves, L. M. Pavelski, C. P. de Almeida, J. N. Kuk, S. M. Venske and M. R. Delgado, "Adaptive operator selection for many-objective optimization with NSGA-III", in 2017: *Evol. Multi-Criterion Optimiz.* vol. 10173, Springer, pp 267-281.

[50] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.,* vol. 13, no. 2, pp. 284-302, 2009.

[51] 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.

[52] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, 2000.

[53] K. Deb, L. Thiele, M. Laumanns and E. Zitzler, "Scalable multi-objective optimization test problems," *Congress Evol. Comput.*, 2002, pp. 825–830.

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

[55] K. Tagawa, H. Shimizu and H. Nakamura, "Indicator-based differential evolution using exclusive hypervolume approximation and parallelization

for multi-core processors," *13th Annual Genetic and Evol. Comput. Conf*., Dublin, Ireland, July 12-16, pp. 657-664, 2011.
[56] J. D. Gibbons, and S. Chakraborti. *Nonparametric Statistical Inference*, 5th Ed., FL: Chapman & Hall/CRC Press, Taylor & Francis Group, 2011.
[57] M. Hollander, and D. A. Wolfe. *Nonparametric Statistical Methods*. Hoboken, NJ: John Wiley & Sons, Inc., 1999.
[58] Q. Zhu, Q. Lin, W. Chen, K. Wong, C. A. Coello Coello, J. Li, J. Chen and J. Zhang, "An external archive-guided multiobjective particle swarm optimization algorithm." *IEEE Trans. Cybern*., 47(9), pp. 2794-2808, 2017.
[59] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in 2006 *IEEE International Conf. on Evol. Comput.*, 2006, pp. 1179-1186.
[60] R. A. Gonçalves, J. N. Kuk, C. P. Almeida and S. M. Venske, "MOEA/D-HH: A hyper-heuristic for multi-objective problems," in *Int. Conf. on Evolut. Multi-Criterion Optimiz.*, 2015, vol. 9018, pp. 94-108.

Wenjun Wang received the Ph.D. degree from Department of Mathematics and System, National University of Defense Technology, Changsha, China, in 2013. He is currently a postdoc research fellow in College of Computer Science and Software Engineering, Shenzhen University.

His current research interests include evolutionary computation, data analysis, systematic models, and machine learning.

Shaoqiang Yang received the B.S. degree from Anyang Normal University, Anyang, China, in 2013, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2017. He is currently a Software Engineer with ZhiHui FinTech limited company, Shenzhen.

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 a Lecturer in College of Computer Science and Software Engineering, Shenzhen University. He has published over 30 research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, dynamic system, and machine learning.

Qingfu Zhang (M'01-SM'06-F'17) received the BSc degree in mathematics from Shanxi University, China in 1984, the MSc degree in applied mathematics and the PhD degree in information engineering from Xidian University, China, in 1991 and 1994, respectively.

He is a Professor at the Department of Computer Science, City University of Hong Kong, Hong Kong. His main research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications.

Dr. Zhang is an Associate Editor of the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Cybernetics. He is also an Editorial Board Member of three other international journals. He is a Web of Science highly cited researcher in Computer Science in 2016, 2017 and 2018.

Ka-Chun Wong received his B.Eng. in Computer Engineering from The Chinese University of Hong Kong in 2008. He has also received his M.Phil. degree at the same university in 2010. He received his PhD degree from the Department of Computer Science, University of Toronto in 2015. After that, he assumed his duty as assistant professor at City University of Hong Kong. His research interests include Bioinformatics, Computational Biology, Evolutionary Computation, Data Mining, Machine Learning, and Interdisciplinary Research.

Carlos A. Coello Coello (M'98-SM'04-F'11) received PhD degree in computer science from Tulane University, USA, in 1996. He is currently Professor (CINVESTAV-3F Researcher) at the Computer Science Department of CINVESTAV-IPN, in Mexico City, México. Dr. Coello 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). Currently, he is associate editor of the *IEEE Transactions on Evolutionary Computation* and serves in the editorial board of 12 other international journals. His major research interests are: evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms. He received 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 a Fellow of the IEEE, and a member of the ACM and the Mexican Academy of Science.

Jianyong Chen (M'11) is a professor in College of Computer Science and Software Engineering, Shenzhen University. He got his PhD from City University of Hong Kong, Hong Kong, China, in 2003. He is interested in Artificial Intelligence and Information Security.

He worked for ZTE Corporation as senior engineer of network technology from 2003 to 2006. After that, he joined the Shenzhen University. He was vice-chairman of International Telecommunication Union-Telecommunication (ITU-T) SG17 from 2004 to 2012, and editor of three recommendations developed in ITU-T SG17. He has published more than 30 papers and got more than 30 patents in the field of Artificial Intelligence and Information Security.