Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms

Aimin Zhou, *Member, IEEE*, and Qingfu Zhang, *Senior Member, IEEE*

***Abstract*—Decomposition-based multiobjective evolutionary algorithms (MOEAs) decompose a multiobjective optimization problem into a set of scalar objective subproblems and solve them in a collaborative way. A naïve way to distribute computational effort is to treat all the subproblems equally and assign the same computational resource to each subproblem. This paper proposes a generalized resource allocation (GRA) strategy for decomposition-based MOEAs by using a probability of improvement vector. Each subproblem is chosen to invest according to this vector. An offline measurement and an online measurement of the subproblem hardness are used to maintain and update this vector. Utility functions are proposed and studied for implementing a reasonable and stable online resource allocation strategy. Extensive experimental studies on the proposed GRA strategy have been conducted.**

***Index Terms*—Decomposition, multiobjective optimization, resource allocation.**

I. Introduction

THIS PAPER considers the following box-constrained multiobjective optimization problem (MOP):

$$\begin{aligned} \min\ & F(x) = (f_1(x), \ldots, f_m(x)) \\ \text{s.t}\ & x \in \Omega \end{aligned} \tag{1}$$

where $\Omega = [a_i, b_i]^n$ is the feasible region of the search (or decision variable) space, and $x = (x_1, \ldots, x_n)$ is a decision variable vector. $F : \Omega \to R^m$ consists of m objective functions $f_i(x)$, $i = 1, \ldots, m$, and R^m is the objective space. Since the objectives often conflict with one another, there does not exist a single solution that can optimize all objectives at the same time. Therefore, a decision maker should seek solutions that can balance different objectives in an optimal way.

Let $x, y \in \Omega$. x is said to dominate y, denoted by $F(x) \prec F(y)$, if and only if $f_i(x) \le f_i(y)$ for every $i \in \{1, \ldots, m\}$ and $F(x) \ne F(y)$.[1] Given a set $S \subseteq \Omega$, a point is called nondominated in S if no other point in S dominates it. A point $x^* \in \Omega$ is Pareto-optimal if it is nondominated in the attainable objective set. $F(x^*)$ is then called a Pareto-optimal (objective) vector. In other words, any improvement in one objective of a Pareto optimal point must lead to deterioration of at least another objective. The set of all the Pareto-optimal points is called the Pareto set and the set of all the Pareto-optimal objective vectors is the Pareto front (PF) [1], [2]. In many real-life applications, the PF is of great interest to a decision maker for understanding the tradeoffs among different objectives and selecting their final solutions.

Multiobjective evolutionary algorithms (MOEAs) have been accepted as a basic tool for approximating the PF of an MOP in a single run [3]. Over the last two decades, three major evolutionary algorithm paradigms have been developed, i.e., the Pareto domination-based approaches (see [4]–[10]), the indicator-based approaches (see [11]–[14]), and the decomposition-based approaches (see [15]–[17]). A multiobjective evolutionary algorithm based on decomposition (MOEA/D) [16], [17] is a decomposition algorithm framework, which generalizes the cellular multiobjective genetic algorithm [15]. It decomposes an MOP into a set of scalar objective subproblems[2] and solves them in a collaborative way. In MOEA/D, each solution in its population is associated with a different subproblem, and its solution replacement and selection operations are mainly based on the subproblem objective function values instead of domination relationships. Recently, a number of MOEA/D variants have been proposed [19]–[24] and applied to various MOPs [25]–[34].

In the original MOEA/D [16], all the subproblems are treated equally and the amount of computational resource assigned to each subproblem is the same. It has been observed, as one can expect, that some parts of the PF in an MOP can be more difficult to approximate than others (see [35]–[37]). To reduce the computational cost, it is very natural to allocate

Manuscript received July 31, 2014; revised December 24, 2014; accepted April 8, 2015. Date of publication April 17, 2015; date of current version January 27, 2016. This work was supported by the National Basic Research Program under Grant 2011CB707104, the China National Instrumentation Program under Grant 2012YQ180132, the National Natural Science Foundation of China under Grant 61273313 and Grant 61473241, and the start-up Grant 7200386 of City University of Hong Kong.

A. Zhou is with Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China, and also with the Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China (e-mail: amzhou@cs.ecnu.edu.cn).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qingfu.zhang@cityu.edu.hk; qzhang@essex.ac.uk).

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

Digital Object Identifier 10.1109/TEVC.2015.2424251

[1] In the case of maximization, the inequality signs should be reversed.

[2] Some MOEA/D variants decompose an MOP into a number of multiobjective subproblems [18].

different computational resources to different subproblems according to their difficulties. Adaptation strategies have been widely used to improve the performance of evolutionary algorithms including MOEAs. These strategies aim to adjust the behavior of an algorithm in an online manner to suit the problem in question (see [38], [39]). Much effort has been made along this direction and it is still attracting a lot of attention in the community of evolutionary computation [40], [41]. All algorithmic components such as representation schemes, control parameters, variation operators and algorithm structures can be tuned adaptively [39], [42]. Feedback information is often needed for adaptation strategies.

This paper continues our work on dynamic computational resource allocation in MOEA/D, which was initialized in [35], and proposes a generalized resource allocation (GRA) strategy. We denote MOEA/D with the GRA strategy as MOEA/D-GRA hereafter. In MOEA/D-GRA, a probability of improvement (PoI) vector is introduced and each subproblem is associated with a different PoI element. At each generation, some subproblems are chosen for investment according to the PoI vector and thus the computational resource can be assigned to these subproblems. This paper systematically studies how to define and maintain the PoI vector. An offline measurement and an online measurement of subproblem hardness are introduced. It demonstrates that online resource allocation (ONRA) is more practical than offline resource allocation (OFRA). Utility functions are also studied for designing an effective and reasonable PoI vector.

The rest of this paper is organized as follows. Section II presents the MOEA/D-GRA framework and its implementation details. Section III introduces an offline strategy and an online strategy for subproblem hardness measurement and the PoI vector definition. Section IV further investigates some other possibilities for ONRA. Section V studies the algorithm parameter settings in MOEA/D-GRA. Section VI compares our approach with the original MOEA/D and a variant with a dynamic resource allocation strategy. Finally, this paper is concluded in Section VII with some future research topics.

II. Algorithm Framework With GRA

MOEA/D decomposes (1) into N scalar objective or multiobjective subproblems. In this paper, we consider scalar objective decomposition. The objective function of each subproblem can be a weighted linear or nonlinear aggregation of all the objective functions. The optimal solution of each subproblem is a different Pareto optimal solution to (1). All these N optimal solutions hopefully constitute a good approximation to the PF of (1). MOEA/D attempts to optimize all the N scalar objective optimization subproblems in a single run. A neighborhood relationship among all the subproblems is defined based on the distances among their weights. A basic assumption behind MOEA/D is that neighboring subproblems should have similar optimal solutions, which is reasonable since two subproblems with close weights have similar objective functions. MOEA/D makes use of the neighborhood relationship for promoting its search efficiency.

Algorithm 1: MOEA/D-GRA Procedure

1 Initialize the weight vectors λ^i, the neighborhood B^i, the solution x^i, p^i for subproblem $i = 1, \ldots, N$.
2 Initialize the ideal point z^* $(j = 1, \ldots, m)$ as

$$z_j^* = \arg \min_{i=1,\ldots,N} f_j(x^i).$$

3 **while** *not terminate* **do**
4   **for** $i = 1$ *to* N **do**
5   **if** $rand() \le p^i$ **then**
6   Generate a trial solution y for subproblem i.
7   Update the ideal point z^* $(j = 1, \ldots, m)$ by

$$z_j^* = \min\{z_j^*, f_j(y)\}.$$

8   Update the population by y.
9   **end**
10   **end**
11   Update the PoI vector $p = (p^1, \ldots, p^N)$.
12 **end**

In this section, we propose a simple yet efficient MOEA/D-GRA. This framework requires the following initial setting.

1) N: The number of subproblems.
2) A decomposition method for decomposing (1) into N subproblems. Suppose that subproblem i is to minimize $g^i(x)$ on Ω.
3) $B^i \subset \{1, \ldots, N\}$: The index set of the neighboring subproblems of subproblem i where $|B^i| = T$.
4) p^i: The probability that subproblem i should be invested.
5) A stopping criterion.

At each generation, MOEA/D-GRA maintains and updates the following data $(i = 1, \ldots, N)$.

1) x^i: The current solution to subproblem i. All the x^i's constitute the current population.
2) $F(x^i)$: The objective function value of x^i.
3) p^i: The probability that subproblem i should be invested.
4) $z^* = (z_1^*, \ldots, z_m^*)$: An estimated ideal point.

The algorithm framework of MOEA/D-GRA is shown in Algorithm 1.

In Algorithm 1, *rand*() returns a uniformly distributed value from (0.0, 1.0). With probability p^i, subproblem i is chosen to be invested on in lines 5–9. The PoI vector p determines how computational resource is allocated among subproblems. In line 1, it is initialized to be $p = (0.5, \ldots, 0.5)$, and in line 11, it is updated based on information collected from the previous search.

A similar dynamic resource allocation strategy has been utilized in MOEA/D with dynamic resource allocation (MOEA/D-DRA) [35], [36] and MOEA/D with adaptive mating selection (MOEA/D-AMS) [37]. In both MOEA/D-DRA and MOEA/D-AMS, some unsolved subproblems according to their utility functions are chosen for evolution in each generation. This could be implemented in MOEA/D-GRA by setting $p^i = 1$ or 0. Therefore, MOEA/D-GRA can be regarded as an extension of MOEA/D-DRA and MOEA/D-AMS. It should also be noted

Algorithm 2: Offspring Reproduction for Subproblem i

1 Set mating pool

$$P = \begin{cases} B^i & \text{if } rand() < \delta \\ \{1, \ldots, N\} & \text{otherwise} \end{cases}.$$

2 Randomly choose two parent indices $r1, r2 \in P$ that are different from each other and different from i.

3 Generate a trial solution

$$y' = x^i + F\left(x^{r1} - x^{r2}\right).$$

4 Repair the trial solution

$$y''_k = \begin{cases} x^i_k - rand()\left(x^i_k - a_k\right) & \text{if } y'_k < a_k \\ x^i_k + rand()\left(b_k - x^i_k\right) & \text{if } y'_k > b_k \\ y'_k & \text{otherwise} \end{cases}$$

for $k = 1, \ldots, n$.

5 Mutate the trial solution

$$y_k = \begin{cases} y'''_k & \text{if } rand() < p_m \text{ and } y'''_k \in [a_k, b_k] \\ y''_k & \text{otherwise.} \end{cases}$$

where $k = 1, \ldots, n$, $y'''_k = y''_k + \Delta_k(b_k - a_k)$. Let $r = rand()$, Δ_k is defined as

$$\Delta_k = \begin{cases} \left[2r + (1-2r)\left(\frac{b_k - y''_k}{b_k - a_k}\right)^{\eta+1}\right]^{\frac{1}{\eta+1}} - 1 & \text{if } r < 0.5 \\ 1 - \left[2 - 2r + (2r-1)\left(\frac{y''_k - a_k}{b_k - a_k}\right)^{\eta+1}\right]^{\frac{1}{\eta+1}} & \text{otherwise.} \end{cases}$$

6 Return $y = (y_1, \ldots, y_n)$.

that in the case of all the $p^i = 1$, this framework becomes the original MOEA/D.

A. Algorithm Specification

The work in this paper is based on MOEA/D-DE [17]. The subproblem definition and implementations of the reproduction and population replacement are as follows.

1) Subproblem Definition: A number of methods could be applied in MOEA/D to convert an MOP into a set of scalar objective subproblems. In this paper, we use the Tchebycheff technique to define a subproblem

$$g\left(x|\lambda^i, z^*\right) = \max_{1\le j\le m} \lambda^i_j \left| f_j(x) - z^*_j \right| \quad (2)$$

where $z^* = (z^*_1, \ldots, z^*_m)$ is an ideal point, and $\lambda^i = (\lambda^i_1, \ldots, \lambda^i_m)$ is a weight vector. Let $w^i = (w^i_1, \ldots, w^i_m)$, $i = 1, \ldots, N$, be a set of evenly distributed vectors that satisfy $\sum_{j=1}^m w^i_j = 1$, and $w^i_j > 0$.[3] The weight vectors are defined as

$$\lambda^i_j = \frac{\frac{1}{w^i_j}}{\sum_{k=1}^m \frac{1}{w^i_k}} \quad (3)$$

where $j = 1, \ldots, m$, and $i = 1, \ldots, N$. In the original MOEA/D, the vectors w^i are used as the weights directly.

[3] If $w^i_j = 0$, it is reset to $w^i_j = 1.0 \times 10^{-5}$ to make division legal in (3).

Algorithm 3: Replacement for Offspring of Subproblem i

1 Find the subproblem that can be improved most by the trial solution y as

$$k = \arg_{j=1,\ldots,N} \max \left\{ \frac{g\left(x^j|\lambda^j, z^*\right) - g\left(y|\lambda^j, z^*\right)}{g\left(x^j|\lambda^j, z^*\right)} \right\}.$$

2 If $g(y|\lambda^k, z^*) < g(x^k|\lambda^k, z^*)$, set $x^k = y$.

Some recent studies show that the setting in (3) will lead to better distribution on the PF [43], [44]. For this reason, we define the weight vectors by (3) for MOEA/D and its variants in this paper. The details on how to generate a set of evenly distributed w vectors can be found in [16].

2) Offspring Reproduction: Line 6 in Algorithm 1 is to generate a new solution y. There are many ways to implement it. For example, one can crossover two randomly selected solutions from $\{x^k|k \in B^i\}$ for producing a new solution, and one can also do scalar objective local search to improve x^i for generating y.

In this paper, we use the same reproduction procedure as in MOEA/D-DE [17] to generate an offspring solution for subproblem i and the details are shown in Algorithm 2. In line 1, a mating pool is set to be the solutions from the neighboring subproblems with a probability δ or the whole population with a probability $1 - \delta$. A trial solution is generated by using the current solution and two parents from the mating pool in lines 2–3. This operator is actually from the differential evolution [45] algorithm. After that, the trial solution is repaired to make it feasible in line 4. Finally, the trial solution is mutated through the polynomial mutation [4] in line 5.

3) Population Replacement: Line 8 in Algorithm 1 may replace one solution in the population by the trial solution y. Unlike Pareto domination-based approaches, MOEA/D and its variants use subproblem objective values to do replacement. In the original MOEA/D [17], the newly generated solution can replace several neighboring solutions of its subproblem. It has been shown that it could be more efficient to replace the neighboring solutions of its best matched subproblem [46]. Following this idea and our previous work [47], we define the best matched subproblem as the one that can be improved most among all the subproblems. The reason is that in our approach, only some of the subproblems are chosen for offspring reproduction, and the offspring solutions might be suitable for some subproblems that are outside of the neighborhoods of the chosen subproblems. Comparing with the original strategy proposed in [17] and [47], the new replacement strategy does not need any control parameter. The details of the replacement procedure are presented in Algorithm 3.

III. Offline and Online Resource Allocation Strategies

A key issue in the implementation of MOEA/D-GRA is how to set and update the PoI vector p, i.e., allocate computational resources to different subproblems. Naturally, there are two strategies for it.

1) *OFRA:* Set p according to a hardness measure of subproblems in advance.
2) *ONRA:* Dynamically adjust p according to a figure of merit of investments on subproblems online.

This section is to implement and compare these two strategies. We consider the following two minimization problems from [17] in our studies:

$$T1\begin{cases} f_1(x) = x_1 + \frac{2}{|J_1|}\sum\limits_{j\in J_1}\left(x_j - \sin\left(6\pi x_1 + \frac{j}{n}\pi\right)\right)\\ f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum\limits_{j\in J_2}\left(x_j - \sin\left(6\pi x_1 + \frac{j}{n}\pi\right)\right)\end{cases}$$

$$T2\begin{cases} f_1(x) = x_1 + \frac{2}{|J_1|}\sum\limits_{j\in J_1}\left(x_j - \frac{4}{5}x_1\sin\left(6\pi x_1 + \frac{j}{n}\pi\right)\right)\\ f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum\limits_{j\in J_2}\left(x_j - \frac{4}{5}x_1\sin\left(6\pi x_1 + \frac{j}{n}\pi\right)\right)\end{cases}$$

where $x \in [0, 1]\times[-1, 1]^{n-1}$, $J_1 = \{j|j$ is odd and $2 \le j \le n\}$, and $J_2 = \{j|j$ is even and $2 \le j \le n\}$.

A. Offline Resource Allocation

To implement this strategy, one should first define and calculate a measure of subproblem hardness, which is a difficult task. Since our purpose is to do a proof of concept research for this strategy, we will not spend effort on developing practical methods for estimating subproblem hardness in this paper. Instead, we run the composite differential evolution (CoDE) [48] on each subproblem and record the average number of function evaluations ($\overline{FE}$) for reducing the objective function value below a given error among 51 independent runs. Then, we use $\overline{FE}$ to measure the hardness of each subproblem in our studies.

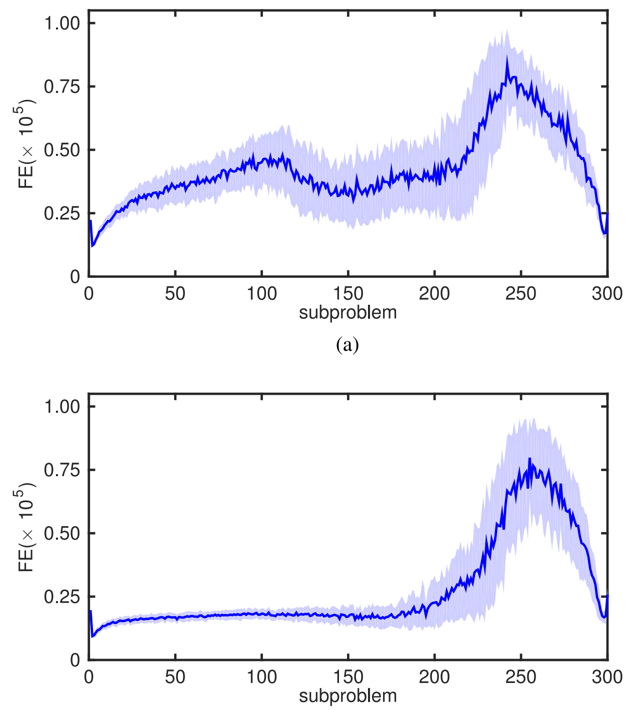
In the experimental studies, both $T1$ and $T2$ are decomposed into 300 subproblems with an ideal point $z^* = (0, 0)$. The population size of CoDE is 30 and the maximum number of FEs is 100 000. Let $g^* = \min g(x)$, define the error of a solution as error $= |g(x) - g^*|$.

Fig. 1 plots the cost of each subproblem to achieve the goal that error $\le 10^{-5}$ over 51 runs. It is clear from Fig. 1 that, to have the same error value, the cost used by each subproblem is different. On $T1$, the maximum and minimum costs to tackle a subproblem are about 8.36×10^4 and 1.23×10^4 FEs, respectively. On $T2$, they are about 7.98×10^4 and 0.94×10^4 FEs, respectively.

Based on the cost used by each subproblem, we define the PoI vector as

$$p^i = \frac{\overline{FE}^i}{\max\limits_{j=1,\ldots,N}\left\{\overline{FE}^j\right\}} \tag{4}$$

where $\overline{FE}^i$ denotes the average cost consumed by CoDE to tackle subproblem i ($i = 1, \ldots, N$). It is clear that with this definition, the computational cost assigned to subproblem i is propositional to p^i, which is propositional to $\overline{FE}^i$.



(a)

(b)

Fig. 1. FE values for different subproblems to achieve the goal that error $\le 10^{-5}$ over 51 runs. The solid line is the average value and the shadow represents the standard deviation. (a) $T1$. (b) $T2$.

B. Online Resource Allocation

To implement this strategy, a key issue is to measure subproblem hardness in an online manner. As in [35], an utility function is defined for this purpose. Let x_t^i and $x_{t-\Delta T}^i$ denote the solutions of subproblem i in the current generation t and the $(t - \Delta T)$th generation, respectively. Define an utility function as the relative improvement in the last ΔT generations as

$$u^i = \frac{g^i\left(x_{t-\Delta T}^i\right) - g^i\left(x_t^i\right)}{g^i\left(x_{t-\Delta T}^i\right)}. \tag{5}$$

Define the PoI vector as

$$p^i = \frac{u^i + \varepsilon}{\max\limits_{j=1,\ldots,N}\left\{u^j\right\} + \varepsilon} \tag{6}$$

where $i = 1, \ldots, N$, $\varepsilon = 1.0 \times 10^{-50}$ is a small value to keep a legal division.

If a subproblem has been improved over the last ΔT generations, it should have a high probability to be improved over the next few generations. By using the above PoI vector, each subproblem has a probability to be chosen and the one that has been improved more over the last ΔT generations has a higher probability to be chosen again. It should also be noticed that once no subproblem can be improved over the last ΔT generations, i.e., $\max\limits_{i=1,\ldots,N}\{u^i\} = 0$, the PoI vectors will be reset, i.e., $p^i = 1$ for $i = 1, \ldots, N$, and all subproblems will be chosen for offspring reproduction.

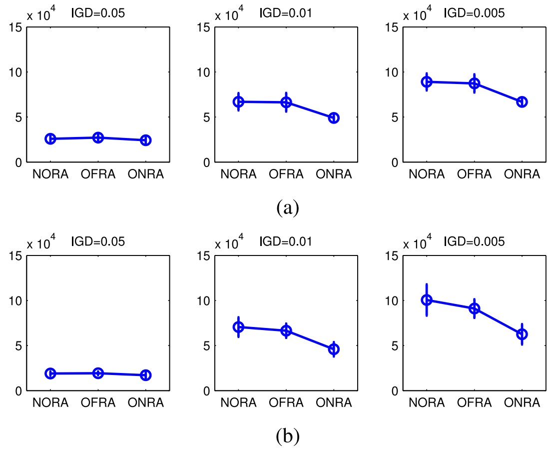


Fig. 2. FE values required by NORA, OFRA, and ONRA to achieve the three IGD levels over 51 runs. (a) $T1$. (b) $T2$.

C. Experimental Results

In this subsection, we empirically study the performance of the resource allocation strategies in Algorithm 1. MOEA/D with no resource allocation (NORA), with OFRA, and with ONRA strategies are compared on $T1$ and $T2$.

The parameters in MOEA/D are: the population size $N = 300$, the neighborhood size $T = 20$. The parameters in offspring reproduction are $\delta = 0.8$, $F = 0.5$, and $\eta = 20$. The maximum FE number is 1.5×10^5 for all the algorithms. In ONRA, $\Delta T = 20$. The following results are based on 51 independent runs. The inverted general distance (IGD) metric, which shall be introduced in detail later in Section VI, is used to measure the solution quality.

1) Comparison Study: Fig. 2 presents the costs to reach below three IGD levels: 0.05, 0.01, and 0.005, with the three strategies on $T1$ and $T2$. Firstly, we compare the NORA and OFRA strategies. It is clear that on $T1$, OFRA and NORA have similar performance. On $T2$, OFRA requires less average FEs to achieve the goals than NORA, and moreover, OFRA is more stable than NORA especially in late stages. This indicates that the fixed PoI somehow reflects the difficulty of the subproblems. Secondly, we compare the ONRA and OFRA strategies. ONRA performs better than OFRA on both $T1$ and $T2$. The reason might be that (a) a subproblem does not evolve independently but cooperatively with its neighboring subproblems in MOEA/D, and (b) in different stages or different runs, the resources needed by each subproblem might be different. The ONRA strategy can successfully detect subproblem difficulties.

2) Behavior of Resource Allocation: Fig. 3 plots the costs for different subproblems over 51 runs. Comparing Figs. 1 and 3, we can see that the cost used by OFRA is very consistent with that in Fig. 1. But it is not the case for ONRA. It could be due to the collaboration among different subproblems in MOEA/D. In different runs and different stages, the resource needed by each subproblem might be different, and hard subproblems can be solved efficiently using collaboration.

In the following, we will conduct more investigation.

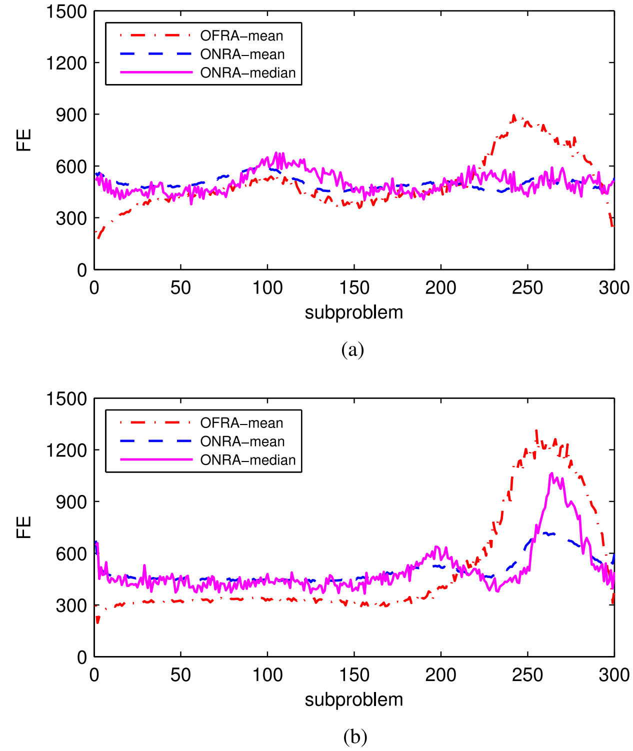


Fig. 3. FE values for different subproblems over 51 runs. The mean cost for OFRA and ONRA, the cost of the median run in terms of the final obtained IGD value over 51 runs for ONRA are plotted. (a) $T1$. (b) $T2$.

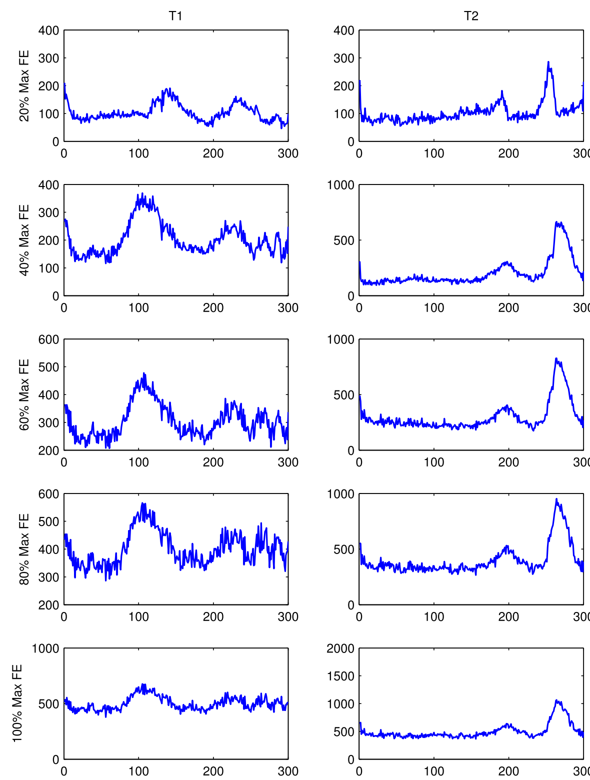
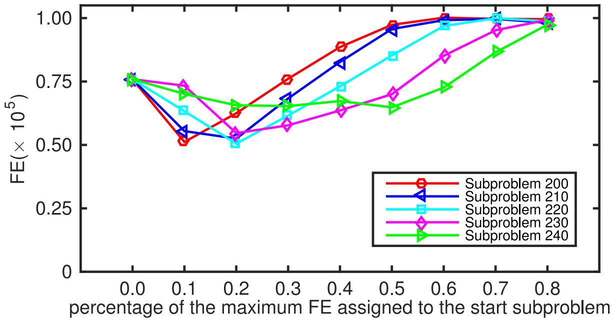
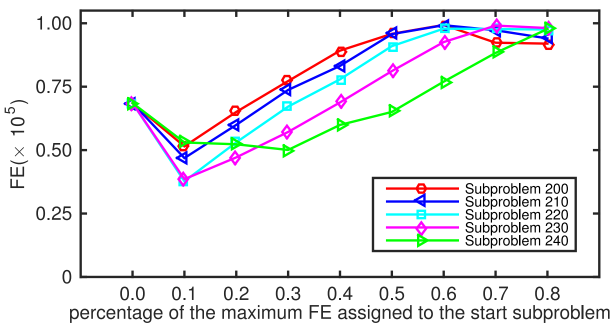


Fig. 4. Cost assigned to each subproblem after 20%, 40%, 60%, 80%, and 100% of the maximum FEs in the median run according to the final IGD values among 51 runs on $T1$ and $T2$, respectively.

Firstly, we consider the cost used in different stages in the run in which the final IGD value is the median among the 51 runs. Fig. 4 plots the cost assigned to each subproblem



(a)



(b)

Fig. 5. Average FE values required to achieve the goal that error $\leq 10^{-5}$ over 51 runs for subproblem 250. The optimization process starts from subproblems 200, 210, 220, 230, and 240 with different proportions of the maximum FE values. (a) $T1$. (b) $T2$.

when five different percentages of the maximum FEs have been used on $T1$ and $T2$, respectively. Take $T1$ as an example. In early stage, the subproblems around subproblem 250 consume more computational budget than the other subproblems, while in the middle and later stages, the subproblems around subproblem 100 receive more computational effort.

Secondly, we investigate the cooperation among neighboring subproblems by the following experiment. Take subproblem 250, which is among the hardest subproblems according to Fig. 1, on both $T1$ and $T2$ as an example. The optimization process starts by optimizing subproblems 200, 210, 220, 230, and 240 in the experiment, and after some time (10%-80% of the maximum FEs) the optimization process turns to optimize subproblem 250 by using the obtained solutions as its initial population. As in Section III-A, CoDE is used as the optimizer. The maximum FE value is 100 000, and the results are based on 51 independent runs. Fig. 5 shows the average FE values required to achieve the goal that error $\leq 10^{-5}$. The figure clearly shows that starting from an easier subproblem could reduce the average FE values required to tackle a hard subproblem. From Fig. 5, we can see that by assigning about 10%-30% maximum FE value to a close and easier subproblem, the total FE value to tackle subproblem 250 is reduced. Fig. 4 also shows that on $T1$, subproblem 250 is no longer the hardest, and on $T2$, except subproblem 250, subproblem 200 also becomes hard.

To visualize the results, Figs. 6 and 7 show the median approximations obtained by the NORA, OFRA, and ONRA strategies after 20%, 40%, and 60% FEs on $T1$ and $T2$, respectively. From these figures, it is clear that ONRA can

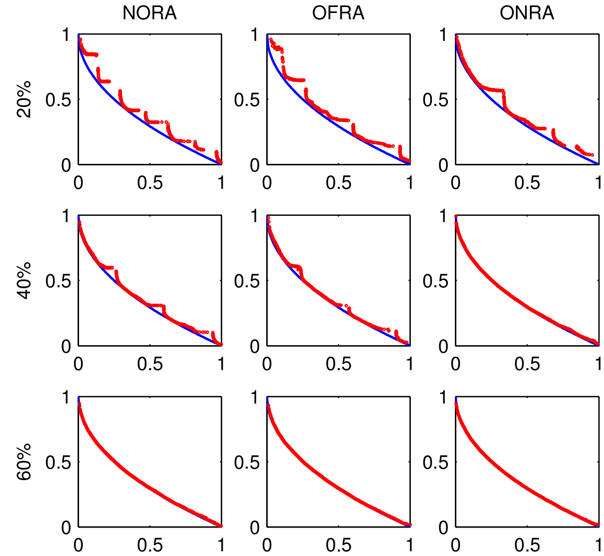


Fig. 6. Median (according to the IGD metric values) approximations obtained by NORA, OFRA, and ONRA after 20%, 40%, and 60% FEs on $T1$.

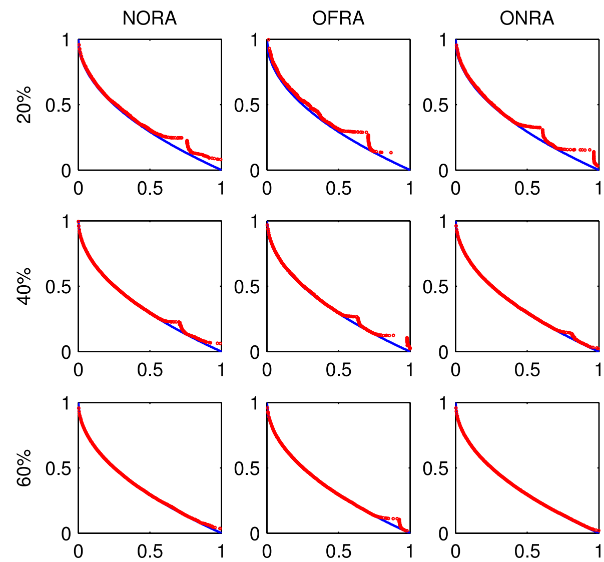


Fig. 7. Median (according to the IGD metric values) approximations obtained by NORA, OFRA, and ONRA after 20%, 40%, and 60% FEs on $T2$.

approximate the PF well after 40% FEs while NORA and OFRA still miss some parts of the PF.

3) Conclusion: From these experiments, we can conclude that (a) MOEA/D with resource allocation works better than the one without resource allocation, (b) the OFRA strategy is not practical for real-world problems since the cost to tackle each subproblem is unknown beforehand, and (c) the ONRA strategy is able to adaptively allocate the computational resources for different problems, and in different running stages.

IV. Alternative Utility Functions

In the above section, we have used the improvement of a subproblem as the utility function. This section considers the following alternative utility functions.

TABLE I

MEAN$_{\text{STD.}}$ IGD VALUES FOR DIFFERENT RESOURCE ALLOCATION STRATEGIES AFTER 20%, 40%, 60%, 80%, AND 100% FEs OVER 51 RUNS. THE NUMBER IN THE SQUARE BRACKET DENOTES THE RANK AMONG THE NINE STRATEGIES

	20%	40%	60%	80%	100%	Average Rank
			T1			
NORA	$0.0407_{0.0076}$[2]	$0.0126_{0.0037}$[2]	$0.0052_{0.0013}$[4]	$0.0033_{0.0002}$[3]	$0.0027_{0.0001}$[2]	2.60
OFRA	$0.0433_{0.0093}$[5]	$0.0127_{0.0046}$[4]	$0.0051_{0.0014}$[2]	$0.0033_{0.0003}$[2]	$0.0027_{0.0001}$[3]	3.20
ONRA	$0.0353_{0.0091}$[1]	$0.0060_{0.0009}$[1]	$0.0035_{0.0003}$[1]	$0.0028_{0.0002}$[1]	$0.0024_{0.0001}$[1]	1.00
ONRA$_x$	$0.0626_{0.0116}$[9]	$0.0397_{0.0091}$[9]	$0.0344_{0.0103}$[9]	$0.0313_{0.0103}$[9]	$0.0292_{0.0092}$[9]	9.00
ONRA$_f$	$0.0501_{0.0120}$[8]	$0.0230_{0.0099}$[8]	$0.0112_{0.0059}$[8]	$0.0060_{0.0037}$[8]	$0.0039_{0.0021}$[8]	8.00
ONRA$_{xf}$	$0.0461_{0.0086}$[7]	$0.0198_{0.0058}$[7]	$0.0105_{0.0047}$[7]	$0.0059_{0.0030}$[7]	$0.0037_{0.0010}$[7]	7.00
ONRA$_{ix}$	$0.0411_{0.0081}$[3]	$0.0127_{0.0048}$[3]	$0.0051_{0.0016}$[3]	$0.0034_{0.0003}$[4]	$0.0028_{0.0002}$[4]	3.40
ONRA$_{if}$	$0.0445_{0.0081}$[6]	$0.0154_{0.0052}$[6]	$0.0063_{0.0026}$[6]	$0.0036_{0.0005}$[6]	$0.0029_{0.0002}$[6]	6.00
ONRA$_{ixf}$	$0.0414_{0.0088}$[4]	$0.0145_{0.0056}$[5]	$0.0060_{0.0025}$[5]	$0.0035_{0.0005}$[5]	$0.0028_{0.0002}$[5]	4.80
			T2			
NORA	$0.0296_{0.0048}$[6]	$0.0129_{0.0033}$[6]	$0.0062_{0.0018}$[5]	$0.0039_{0.0012}$[6]	$0.0029_{0.0005}$[6]	5.80
OFRA	$0.0305_{0.0044}$[8]	$0.0124_{0.0032}$[4]	$0.0053_{0.0014}$[2]	$0.0030_{0.0003}$[2]	$0.0024_{0.0001}$[2]	3.60
ONRA	$0.0229_{0.0067}$[1]	$0.0052_{0.0016}$[1]	$0.0033_{0.0008}$[1]	$0.0026_{0.0003}$[1]	$0.0023_{0.0003}$[1]	1.00
ONRA$_x$	$0.0489_{0.0077}$[9]	$0.0374_{0.0091}$[9]	$0.0331_{0.0079}$[9]	$0.0319_{0.0077}$[9]	$0.0309_{0.0070}$[9]	9.00
ONRA$_f$	$0.0300_{0.0047}$[7]	$0.0156_{0.0048}$[8]	$0.0094_{0.0039}$[8]	$0.0056_{0.0038}$[8]	$0.0034_{0.0012}$[8]	7.80
ONRA$_{xf}$	$0.0286_{0.0036}$[4]	$0.0133_{0.0033}$[7]	$0.0067_{0.0021}$[7]	$0.0039_{0.0010}$[7]	$0.0029_{0.0006}$[5]	6.00
ONRA$_{ix}$	$0.0276_{0.0036}$[3]	$0.0112_{0.0027}$[2]	$0.0057_{0.0014}$[3]	$0.0037_{0.0010}$[3]	$0.0028_{0.0005}$[4]	3.00
ONRA$_{if}$	$0.0294_{0.0043}$[5]	$0.0126_{0.0035}$[5]	$0.0064_{0.0021}$[6]	$0.0037_{0.0009}$[5]	$0.0028_{0.0003}$[3]	4.80
ONRA$_{ixf}$	$0.0271_{0.0037}$[2]	$0.0116_{0.0043}$[3]	$0.0057_{0.0019}$[4]	$0.0037_{0.0014}$[4]	$0.0029_{0.0013}$[7]	4.00

A. Density in Objective Space

This utility function considers the population distribution in the objective space. We use the volume covered by the neighboring solutions to denote the sparseness of subproblem i. Since our target is to find a well distributed approximation to the PF, solutions in sparse areas should receive more attention

$$u_f^i = \prod_{j=1}^{m}\left(\max_{k\in B^i}\left\{f_j\left(x^k\right)\right\} - \min_{k\in B^i}\left\{f_j\left(x^k\right)\right\}\right) \tag{7}$$

where $i = 1, \dots, N$.

B. Density in Decision Space

This utility function is the same as the one above, and the only difference is that this strategy considers the population distribution in the decision space

$$u_x^i = \prod_{j=1}^{n}\left(\max_{k\in B^i}\left\{x_j^k\right\} - \min_{k\in B^i}\left\{x_j^k\right\}\right) \tag{8}$$

where $i = 1, \dots, N$.

C. Hybrid Utility Functions

These utility functions consider more than one issue at the same time. We propose the following functions by combining u, u_f, and u_x:

$$u_{xf}^i = \frac{u_f^i + \varepsilon}{\max_j\left\{u_f^j\right\} + \varepsilon} + \frac{u_x^i + \varepsilon}{\max_j\left\{u_x^j\right\} + \varepsilon} \tag{9}$$

$$u_{ix}^i = \frac{u^i + \varepsilon}{\max_j\left\{u^j\right\} + \varepsilon} + \frac{u_x^i + \varepsilon}{\max_j\left\{u_x^j\right\} + \varepsilon} \tag{10}$$

$$u_{if}^i = \frac{u^i + \varepsilon}{\max_j\left\{u^j\right\} + \varepsilon} + \frac{u_f^i + \varepsilon}{\max_j\left\{u_f^j\right\} + \varepsilon} \tag{11}$$

$$u_{ixf}^i = \frac{u^i + \varepsilon}{\max_j\{u^j\} + \varepsilon} + \frac{u_x^i + \varepsilon}{\max_j\left\{u_x^j\right\} + \varepsilon} + \frac{u_f^i + \varepsilon}{\max_j\left\{u_f^j\right\} + \varepsilon} \tag{12}$$

where $i = 1, \dots, N$.

D. Experimental Results

The six utility functions are used in (6) of the ONRA strategy to substitute (5). We denote ONRA with u_x, u_f, u_{xf}, u_{ix}, u_{if}, and u_{ixf} as ONRA$_x$, ONRA$_f$, ONRA$_{xf}$, ONRA$_{ix}$, ONRA$_{if}$, and ONRA$_{ixf}$ strategies, respectively.

In this section, we compare all the resource allocation strategies introduced in this paper on $T1$ and $T2$. Table I presents the statistical results for all the nine resource allocation strategies in MOEA/D-GRA. The mean and std. IGD values are recorded after 20%, 40%, 60%, 80%, and 100% FEs over 51 runs. Table II shows the mean (std.) numbers of FEs ($\times 10^5$) for the nine resource allocation strategies to achieve IGD values of 0.1, 0.05, 0.01, and 0.005 over 51 runs. The light gray numbers and dark gray ones denote the best and worst mean values, respectively. The number of successful runs and the rank of each strategy are also presented.

Firstly, we compare the newly proposed six utility functions. The results in Tables I and II show that on both $T1$ and $T2$, except for the u_x utility function, all the other utility functions work well. ONRA with u_x cannot achieve the goal that IGD ≤ 0.01. The reason might be that u_x is defined in the decision space of high dimension and this makes u_x unstable. It should be noted that the four hybrid utility functions work better than u_x and u_f for the reason that hybrid utility functions consider more issues and are more stable than u_x and u_f.

Secondly, we compare the seven ONRA strategies. On one hand, the results indicate that ONRA with the relative improvement u always achieves the best results and it converges much faster than the other six strategies. On the other hand, the hybrid strategies with u also work better than those without it. This suggests that the relative improvement u can successfully detect the search behavior online.

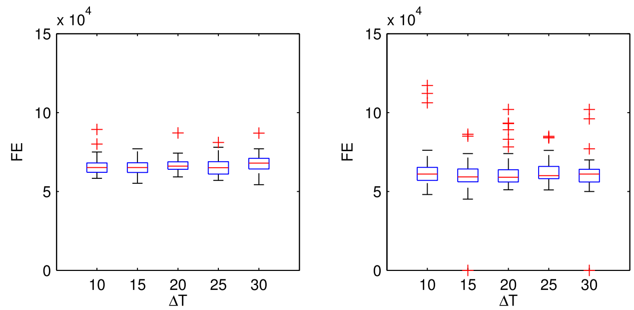
We conclude it is apparent that the adaptive resource allocation can save the computational resource in MOEA/D on the test instances.

V. PARAMETER STUDY

The ONRA strategies do not work without any cost. One additional control parameter, the history length ΔT, is used.

TABLE II
MEAN$_{\text{STD.}}$ NUMBERS OF FE ($\times 10^5$) FOR DIFFERENT RESOURCE ALLOCATION STRATEGIES TO ACHIEVE IGD VALUES OF 0.1, 0.05, 0.01, AND 0.001 OVER 51 RUNS. THE NUMBER IN THE PARENTHESES IS THE NUMBER OF SUCCESSFUL RUNS TO ACHIEVE THE GOAL, AND THE NUMBER IN THE SQUARE BRACKET DENOTES THE RANK AMONG THE 9 STRATEGIES

	0.1	0.05	0.01	0.005	Average Rank
			T1		
NORA	$0.1429_{0.0180}$(51)[2]	$0.2586_{0.0385}$(51)[2]	$0.6698_{0.0954}$(51)[4]	$0.8901_{0.0936}$(51)[4]	3.00
OFRA	$0.1505_{0.0224}$(51)[6]	$0.2720_{0.0430}$(51)[5]	$0.6641_{0.1020}$(51)[3]	$0.8735_{0.1006}$(51)[3]	4.25
ONRA	$0.1358_{0.0210}$(51)[1]	$0.2419_{0.0400}$(51)[1]	$0.4906_{0.0433}$(51)[1]	$0.6677_{0.0455}$(51)[1]	1.00
ONRA$_x$	$0.1936_{0.0362}$(51)[9]	$0.4155_{0.1980}$(49)[9]	NA	NA	9.00
ONRA$_f$	$0.1606_{0.0293}$(51)[8]	$0.3157_{0.0932}$(51)[8]	$0.8821_{0.2579}$(49)[7]	$1.0186_{0.3658}$(46)[8]	7.75
ONRA$_{xf}$	$0.1553_{0.0226}$(51)[7]	$0.2852_{0.0469}$(51)[7]	$0.9289_{0.2206}$(51)[8]	$1.0002_{0.3974}$(45)[7]	7.25
ONRA$_{ix}$	$0.1465_{0.0192}$(51)[4]	$0.2670_{0.0452}$(51)[4]	$0.6593_{0.1084}$(51)[2]	$0.8707_{0.0935}$(51)[2]	3.00
ONRA$_{if}$	$0.1480_{0.0191}$(51)[5]	$0.2798_{0.0433}$(51)[6]	$0.7261_{0.1174}$(51)[6]	$0.9519_{0.1062}$(51)[6]	5.75
ONRA$_{ixf}$	$0.1460_{0.0210}$(51)[3]	$0.2625_{0.0420}$(51)[3]	$0.7096_{0.1317}$(51)[5]	$0.9211_{0.1220}$(51)[5]	4.00
			T2		
NORA	$0.1130_{0.0103}$(51)[8]	$0.1895_{0.0237}$(51)[6]	$0.7054_{0.1084}$(51)[6]	$1.0054_{0.1736}$(51)[7]	6.75
OFRA	$0.1108_{0.0101}$(51)[7]	$0.1926_{0.0195}$(51)[8]	$0.6647_{0.0792}$(51)[4]	$0.9105_{0.1022}$(51)[3]	5.50
ONRA	$0.0995_{0.0092}$(51)[1]	$0.1708_{0.0396}$(51)[1]	$0.4590_{0.0790}$(51)[1]	$0.6249_{0.1133}$(51)[1]	1.00
ONRA$_x$	$0.1258_{0.0092}$(51)[9]	$0.3487_{0.1957}$(51)[9]	NA	NA	9.00
ONRA$_f$	$0.1028_{0.0091}$(51)[2]	$0.1874_{0.0218}$(51)[5]	$0.8544_{0.1893}$(51)[8]	$1.0600_{0.3585}$(47)[8]	5.75
ONRA$_{xf}$	$0.1059_{0.0078}$(51)[3]	$0.1850_{0.0125}$(51)[3]	$0.7283_{0.1123}$(51)[7]	$1.0003_{0.2203}$(50)[6]	4.75
ONRA$_{ix}$	$0.1100_{0.0091}$(51)[5]	$0.1862_{0.0188}$(51)[4]	$0.6500_{0.0910}$(51)[3]	$0.9464_{0.1998}$(50)[4]	4.00
ONRA$_{if}$	$0.1106_{0.0108}$(51)[6]	$0.1916_{0.0199}$(51)[7]	$0.6989_{0.1180}$(51)[5]	$0.9901_{0.1495}$(51)[5]	5.75
ONRA$_{ixf}$	$0.1064_{0.0076}$(51)[4]	$0.1819_{0.0164}$(51)[2]	$0.6317_{0.1291}$(50)[2]	$0.9089_{0.2363}$(49)[2]	2.50



(a) (b)

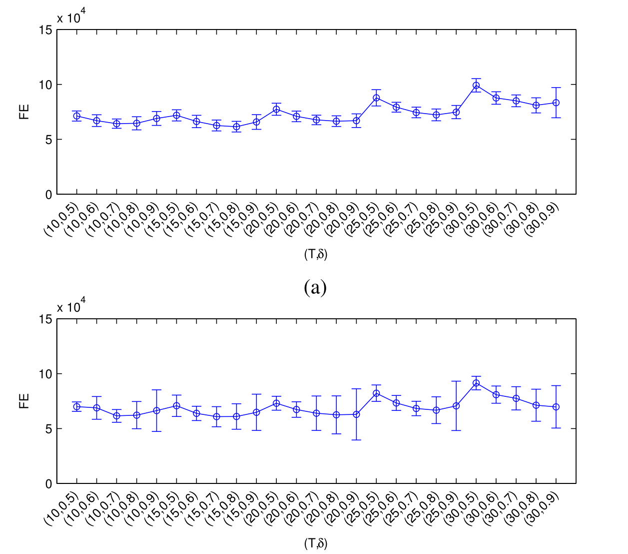
Fig. 8. FE values required by MOEA/D-GRA with different ΔT values to achieve the IGD value of 0.05 over 51 runs. (a) $T1$. (b) $T2$.

In our implementation of MOEA/D, there are also several control parameters: N, T, and δ. In the above sections, according to our experience these parameters are as follows: $N = 300$, $T = 20$, $\delta = 0.8$, and $\Delta T = 20$. Among them, N is the same as in [35], and the setting of T and δ are different from that used in [17]. We use MOEA/D-GRA to denote MOEA/D-GRA with the ONRA strategy hereafter.

In this section, we empirically study the influence of ΔT, T, and δ. The three parameters are classified into two groups: the new parameter ΔT, and the old ones T and δ. In the following, we consider the two groups one by one. MOEA/D-GRA is studied on the two benchmark problems $T1$ and $T2$ with different parameter settings. The other algorithm parameters are the same as in the last paragraph. MOEA/D-GRA with each parameter setting is executed 51 times independently and the algorithm stops after 1.5×10^5 FEs.

A. Influence of ΔT

In the experiments, ΔT is set to 10, 15, 20, 25, and 30. Fig. 8 presents the box plots of the FE values required by MOEA/D-GRA with different control parameters to achieve the IGD value of 0.05 over 51 runs. The experimental results



(a)

(b)

Fig. 9. Average FE values required by MOEA/D-GRA with different control parameters, (T, δ), to achieve the IGD value of 0.05 over 51 runs. (a) $T1$. (b) $T2$.

suggest that although for several parameters there are several outliers, the FE values with different control parameters are relatively stable. The setting of ΔT does not influence the results much.

B. Influence of T and δ

In the experiments, T is set to 10, 15, 20, 25, and 30, and δ is set to 0.5, 0.6, 0.7, 0.8, and 0.9.

Fig. 9 plots the average and standard deviations of the FE values required by MOEA/D-GRA with different control parameters to achieve the IGD value of 0.05 over 51 runs on $T1$ and $T2$, respectively. Fig. 9 shows that: 1) for δ, MOEA/D-GRA with small δ values such as 0.5 or with big ones such as 0.9 does not work well as those with

TABLE III
MEAN$_{\text{STD.}}$ IGD VALUES FOR DIFFERENT RESOURCE ALLOCATION STRATEGIES AFTER 20%, 40%, 60%, 80%, AND 100% FEs OVER 51 RUNS. THE "−," "+," AND "∼" DENOTES THE CORRESPONDING VALUE IS WORSE, BETTER, OR SIMILAR TO THAT OF THE GRA STRATEGY ACCORDING TO THE WILCOXON TEST WITH THE 5% SIGNIFICANCE LEVEL

		20%	40%	60%	80%	100%
F1	DE	$0.0024_{0.0002}(-)$	$0.0016_{0.0000}(-)$	$0.0014_{0.0000}(-)$	$0.0014_{0.0000}(-)$	$0.0013_{0.0000}(-)$
	DRA	$0.0022_{0.0001}(-)$	$0.0015_{0.0000}(-)$	$0.0014_{0.0000}(-)$	$0.0014_{0.0000}(-)$	$0.0013_{0.0000}(-)$
	GRA	$\mathbf{0.0018_{0.0001}}$	$\mathbf{0.0014_{0.0000}}$	$\mathbf{0.0013_{0.0000}}$	$\mathbf{0.0013_{0.0000}}$	$\mathbf{0.0013_{0.0000}}$
F2	DE	$0.0615_{0.0270}(-)$	$0.0223_{0.0101}(-)$	$0.0086_{0.0038}(-)$	$0.0048_{0.0009}(-)$	$0.0039_{0.0005}(-)$
	DRA	$0.0652_{0.0249}(-)$	$0.0228_{0.0155}(-)$	$0.0087_{0.0113}(-)$	$0.0053_{0.0064}(-)$	$0.0043_{0.0059}(-)$
	GRA	$\mathbf{0.0338_{0.0077}}$	$\mathbf{0.0057_{0.0008}}$	$\mathbf{0.0035_{0.0002}}$	$\mathbf{0.0028_{0.0002}}$	$\mathbf{0.0024_{0.0001}}$
F3	DE	$0.0428_{0.0287}(-)$	$0.0264_{0.0241}(-)$	$0.0209_{0.0255}(-)$	$0.0166_{0.0235}(-)$	$0.0122_{0.0198}(-)$
	DRA	$0.0486_{0.0275}(-)$	$0.0311_{0.0282}(-)$	$0.0187_{0.0208}(-)$	$0.0117_{0.0153}(-)$	$0.0081_{0.0135}(-)$
	GRA	$\mathbf{0.0221_{0.0035}}$	$\mathbf{0.0058_{0.0023}}$	$\mathbf{0.0031_{0.0005}}$	$\mathbf{0.0025_{0.0002}}$	$\mathbf{0.0022_{0.0002}}$
F4	DE	$0.0388_{0.0168}(-)$	$0.0217_{0.0163}(-)$	$0.0145_{0.0124}(-)$	$0.0097_{0.0094}(-)$	$0.0072_{0.0087}(-)$
	DRA	$0.0494_{0.0237}(-)$	$0.0289_{0.0224}(-)$	$0.0177_{0.0193}(-)$	$0.0118_{0.0159}(-)$	$0.0074_{0.0118}(-)$
	GRA	$\mathbf{0.0255_{0.0040}}$	$\mathbf{0.0046_{0.0005}}$	$\mathbf{0.0030_{0.0004}}$	$\mathbf{0.0025_{0.0004}}$	$\mathbf{0.0022_{0.0003}}$
F5	DE	$0.0274_{0.0044}(-)$	$0.0195_{0.0050}(-)$	$0.0150_{0.0042}(-)$	$0.0126_{0.0025}(-)$	$0.0111_{0.0023}(-)$
	DRA	$0.0330_{0.0169}(-)$	$0.0207_{0.0143}(-)$	$0.0150_{0.0091}(-)$	$0.0116_{0.0058}(-)$	$0.0099_{0.0060}(-)$
	GRA	$\mathbf{0.0232_{0.0038}}$	$\mathbf{0.0132_{0.0030}}$	$\mathbf{0.0084_{0.0014}}$	$\mathbf{0.0060_{0.0010}}$	$\mathbf{0.0050_{0.0008}}$
F6	DE	$0.0722_{0.0121}(+)$	$\mathbf{0.0513_{0.0050}}(+)$	$\mathbf{0.0471_{0.0060}}(+)$	$0.0433_{0.0060}(+)$	$0.0402_{0.0063}(\sim)$
	DRA	$\mathbf{0.0693_{0.0270}}(+)$	$0.0521_{0.0085}(+)$	$0.0474_{0.0095}(+)$	$\mathbf{0.0428_{0.0105}}(+)$	$\mathbf{0.0385_{0.0105}}(+)$
	GRA	$0.0826_{0.0087}$	$0.0681_{0.0092}$	$0.0593_{0.0133}$	$0.0510_{0.0138}$	$0.0459_{0.0146}$
F7	DE	$0.4537_{0.1212}(-)$	$0.3488_{0.0882}(-)$	$0.2743_{0.1061}(-)$	$0.2044_{0.1264}(-)$	$0.1573_{0.1210}(-)$
	DRA	$0.4378_{0.0820}(-)$	$0.3611_{0.0660}(-)$	$0.3162_{0.0509}(-)$	$0.2785_{0.0448}(-)$	$0.2494_{0.0522}(-)$
	GRA	$\mathbf{0.3903_{0.1438}}$	$\mathbf{0.2397_{0.1053}}$	$\mathbf{0.0914_{0.0727}}$	$\mathbf{0.0258_{0.0274}}$	$\mathbf{0.0075_{0.0080}}$
F8	DE	$0.1565_{0.0319}(-)$	$0.0582_{0.0303}(-)$	$0.0350_{0.0248}(-)$	$0.0233_{0.0207}(-)$	$0.0170_{0.0181}(-)$
	DRA	$0.1312_{0.0373}(-)$	$0.0753_{0.0355}(-)$	$0.0515_{0.0317}(-)$	$0.0365_{0.0286}(-)$	$0.0266_{0.0244}(-)$
	GRA	$\mathbf{0.1018_{0.0255}}$	$\mathbf{0.0356_{0.0231}}$	$\mathbf{0.0159_{0.0112}}$	$\mathbf{0.0071_{0.0042}}$	$\mathbf{0.0042_{0.0022}}$
F9	DE	$0.0769_{0.0351}(-)$	$0.0403_{0.0273}(-)$	$0.0169_{0.0120}(-)$	$0.0075_{0.0029}(-)$	$0.0048_{0.0013}(-)$
	DRA	$0.0823_{0.0321}(-)$	$0.0262_{0.0121}(-)$	$0.0082_{0.0032}(-)$	$0.0047_{0.0011}(-)$	$0.0036_{0.0005}(-)$
	GRA	$\mathbf{0.0395_{0.0199}}$	$\mathbf{0.0068_{0.0020}}$	$\mathbf{0.0036_{0.0005}}$	$\mathbf{0.0027_{0.0003}}$	$\mathbf{0.0023_{0.0002}}$
UF1	DE	$0.0264_{0.0109}(-)$	$0.0039_{0.0016}(-)$	$0.0018_{0.0002}(-)$	$0.0014_{0.0001}(-)$	$0.0012_{0.0001}(-)$
	DRA	$0.0241_{0.0149}(-)$	$0.0047_{0.0037}(-)$	$0.0026_{0.0004}(-)$	$0.0020_{0.0002}(-)$	$0.0017_{0.0002}(-)$
	GRA	$\mathbf{0.0132_{0.0053}}$	$\mathbf{0.0019_{0.0001}}$	$\mathbf{0.0012_{0.0001}}$	$\mathbf{0.0010_{0.0000}}$	$\mathbf{0.0009_{0.0000}}$
UF2	DE	$0.0180_{0.0036}(\sim)$	$0.0117_{0.0035}(-)$	$0.0084_{0.0026}(-)$	$0.0065_{0.0021}(-)$	$0.0052_{0.0017}(-)$
	DRA	$0.0224_{0.0148}(\sim)$	$0.0113_{0.0043}(-)$	$0.0076_{0.0027}(-)$	$0.0053_{0.0017}(-)$	$0.0042_{0.0015}(-)$
	GRA	$\mathbf{0.0175_{0.0030}}$	$\mathbf{0.0059_{0.0015}}$	$\mathbf{0.0025_{0.0004}}$	$\mathbf{0.0019_{0.0005}}$	$\mathbf{0.0016_{0.0003}}$
UF3	DE	$0.0956_{0.0218}(-)$	$0.0278_{0.0153}(-)$	$0.0124_{0.0098}(-)$	$0.0069_{0.0064}(-)$	$0.0039_{0.0028}(-)$
	DRA	$0.0773_{0.0333}(\sim)$	$0.0394_{0.0286}(-)$	$0.0182_{0.0182}(-)$	$0.0080_{0.0108}(-)$	$0.0042_{0.0059}(-)$
	GRA	$\mathbf{0.0709_{0.0314}}$	$\mathbf{0.0215_{0.0177}}$	$\mathbf{0.0070_{0.0054}}$	$\mathbf{0.0024_{0.0013}}$	$\mathbf{0.0014_{0.0004}}$
UF4	DE	$0.0743_{0.0048}(-)$	$0.0654_{0.0041}(-)$	$0.0612_{0.0038}(-)$	$0.0585_{0.0037}(-)$	$0.0565_{0.0036}(-)$
	DRA	$0.0764_{0.0062}(-)$	$0.0689_{0.0058}(-)$	$0.0645_{0.0055}(-)$	$0.0617_{0.0048}(-)$	$0.0595_{0.0043}(-)$
	GRA	$\mathbf{0.0602_{0.0048}}$	$\mathbf{0.0538_{0.0035}}$	$\mathbf{0.0513_{0.0031}}$	$\mathbf{0.0495_{0.0027}}$	$\mathbf{0.0484_{0.0025}}$
UF5	DE	$0.7154_{0.1135}(\sim)$	$0.4852_{0.0846}(-)$	$0.3788_{0.0631}(-)$	$0.3159_{0.0505}(-)$	$0.2860_{0.0452}(-)$
	DRA	$0.7126_{0.1430}(\sim)$	$0.4597_{0.0835}(-)$	$0.3722_{0.0849}(-)$	$0.3279_{0.1033}(-)$	$0.2951_{0.0975}(-)$
	GRA	$\mathbf{0.6917_{0.1017}}$	$\mathbf{0.3723_{0.0565}}$	$\mathbf{0.2690_{0.0362}}$	$\mathbf{0.2400_{0.0245}}$	$\mathbf{0.2335_{0.0289}}$
UF6	DE	$0.3215_{0.0673}(\sim)$	$0.1406_{0.0996}(-)$	$0.1282_{0.1035}(\sim)$	$0.1246_{0.1028}(\sim)$	$0.1227_{0.1021}(-)$
	DRA	$0.3137_{0.2086}(-)$	$0.3006_{0.2312}(-)$	$0.2985_{0.2298}(-)$	$0.2983_{0.2303}(-)$	$0.2977_{0.2401}(-)$
	GRA	$\mathbf{0.3025_{0.0328}}$	$\mathbf{0.1040_{0.0454}}$	$\mathbf{0.0964_{0.0457}}$	$\mathbf{0.0922_{0.0461}}$	$\mathbf{0.0873_{0.0451}}$
UF7	DE	$0.0206_{0.0091}(-)$	$0.0080_{0.0030}(-)$	$0.0039_{0.0013}(-)$	$0.0024_{0.0006}(-)$	$0.0018_{0.0003}(-)$
	DRA	$0.0196_{0.0376}(-)$	$0.0058_{0.0036}(-)$	$0.0036_{0.0030}(-)$	$0.0030_{0.0031}(-)$	$0.0026_{0.0031}(-)$
	GRA	$\mathbf{0.0098_{0.0024}}$	$\mathbf{0.0023_{0.0002}}$	$\mathbf{0.0014_{0.0001}}$	$\mathbf{0.0012_{0.0001}}$	$\mathbf{0.0010_{0.0000}}$
UF8	DE	$0.1483_{0.0207}(-)$	$\mathbf{0.0731_{0.0040}}(+)$	$\mathbf{0.0624_{0.0050}}(+)$	$\mathbf{0.0555_{0.0052}}(+)$	$\mathbf{0.0493_{0.0058}}(\sim)$
	DRA	$\mathbf{0.0922_{0.0284}}(+)$	$0.0752_{0.0156}(+)$	$0.0668_{0.0120}(+)$	$0.0602_{0.0129}(\sim)$	$0.0530_{0.0126}(\sim)$
	GRA	$0.1048_{0.0089}$	$0.0873_{0.0110}$	$0.0766_{0.0146}$	$0.0668_{0.0185}$	$0.0593_{0.0195}$
UF9	DE	$0.2101_{0.0376}(-)$	$0.0783_{0.0298}(-)$	$0.0460_{0.0199}(-)$	$0.0347_{0.0165}(-)$	$0.0272_{0.0076}(-)$
	DRA	$0.1410_{0.0388}(-)$	$0.1229_{0.0466}(-)$	$0.1062_{0.0491}(-)$	$0.0964_{0.0536}(-)$	$0.0891_{0.0564}(-)$
	GRA	$\mathbf{0.0727_{0.0322}}$	$\mathbf{0.0372_{0.0274}}$	$\mathbf{0.0311_{0.0283}}$	$\mathbf{0.0269_{0.0240}}$	$\mathbf{0.0257_{0.0236}}$
UF10	DE	$2.7219_{0.1567}(+)$	$1.9925_{0.2169}(+)$	$1.3794_{0.1237}(+)$	$1.0321_{0.1120}(+)$	$0.8336_{0.0863}(+)$
	DRA	$\mathbf{0.9336_{0.1845}}(+)$	$\mathbf{0.6648_{0.1104}}(+)$	$\mathbf{0.5392_{0.0723}}(+)$	$\mathbf{0.4692_{0.0587}}(+)$	$\mathbf{0.4340_{0.0625}}(+)$
	GRA	$2.8163_{0.2871}$	$2.2577_{0.3285}$	$1.7022_{0.3775}$	$1.3062_{0.3554}$	$1.0356_{0.2841}$

$0.7 \leq \delta \leq 0.8$, and 2) for T, MOEA/D-GRA with small T values works better than those with big ones. It should be noted that MOEA/D-GRA with all these given control parameters can achieve the goal within 1.0×10^5 FEs which is much less than the given maximum FE. Actually, the parameters T and δ control the global search and local search in MOEA/D. It is reasonable that for different problems, the search behaviors might be different. How to adaptively tune these parameters in MOEA/D is worth further investigation.

VI. COMPARISON STUDY

From Section II–V, we have proposed MOEA/D algorithms with different resource allocation strategies. In this section, we compare the proposed best approach, MOEA/D-GRA, with the original MOEA/D-DE [17], and a variant with dynamic resource allocation strategy, MOEA/D-DRA [35]. The three algorithms are applied to 19 benchmark problems. Among them, F1-F9 are from [35], and UF1-UF10 are from [49].

We use the inverted general distance (IGD) [50] and hypervolume difference (I_H^-) [51] metrics to measure the algorithm performance. Let F^* be a nadir point, P^* be an ideal reference PF, and P be the obtained approximation set, the two metrics are defined as

$$\mathrm{IGD}\big(P^*, P\big) = \frac{1}{|P^*|} \sum_{u \in P^*} \min_{v \in P} ||u - v||^2$$

TABLE IV
MEAN$_{\text{STD.}}$ I_H^- VALUES FOR DIFFERENT RESOURCE ALLOCATION STRATEGIES AFTER 20%, 40%, 60%, 80%, AND 100% FEs OVER 51 RUNS. THE "−," "+," AND "∼" DENOTES THE CORRESPONDING VALUE IS WORSE, BETTER, OR SIMILAR TO THAT OF THE GRA STRATEGY ACCORDING TO THE WILCOXON TEST WITH THE 5% SIGNIFICANCE LEVEL

		20%	40%	60%	80%	100%
F1	DE	$0.0041_{0.0003}(-)$	$0.0026_{0.0001}(-)$	$0.0022_{0.0000}(-)$	$0.0020_{0.0000}(-)$	$0.0019_{0.0000}(-)$
	DRA	$0.0038_{0.0002}(-)$	$0.0024_{0.0001}(-)$	$0.0022_{0.0001}(-)$	$0.0021_{0.0000}(-)$	$0.0020_{0.0000}(-)$
	GRA	$\mathbf{0.0032_{0.0002}}$	$\mathbf{0.0022_{0.0000}}$	$\mathbf{0.0020_{0.0000}}$	$\mathbf{0.0019_{0.0000}}$	$\mathbf{0.0018_{0.0000}}$
F2	DE	$0.1226_{0.0499}(-)$	$0.0468_{0.0207}(-)$	$0.0214_{0.0112}(-)$	$0.0128_{0.0041}(-)$	$0.0099_{0.0022}(-)$
	DRA	$0.1331_{0.0502}(-)$	$0.0474_{0.0282}(-)$	$0.0194_{0.0173}(-)$	$0.0123_{0.0119}(-)$	$0.0100_{0.0113}(-)$
	GRA	$\mathbf{0.0632_{0.0149}}$	$\mathbf{0.0134_{0.0025}}$	$\mathbf{0.0086_{0.0011}}$	$\mathbf{0.0070_{0.0008}}$	$\mathbf{0.0061_{0.0006}}$
F3	DE	$0.0675_{0.0286}(-)$	$0.0435_{0.0309}(-)$	$0.0339_{0.0323}(-)$	$0.0277_{0.0310}(-)$	$0.0212_{0.0272}(-)$
	DRA	$0.0824_{0.0396}(-)$	$0.0536_{0.0423}(-)$	$0.0349_{0.0362}(-)$	$0.0231_{0.0281}(-)$	$0.0153_{0.0210}(-)$
	GRA	$\mathbf{0.0407_{0.0060}}$	$\mathbf{0.0130_{0.0058}}$	$\mathbf{0.0073_{0.0022}}$	$\mathbf{0.0058_{0.0012}}$	$\mathbf{0.0050_{0.0010}}$
F4	DE	$0.0672_{0.0204}(-)$	$0.0377_{0.0209}(-)$	$0.0272_{0.0203}(-)$	$0.0190_{0.0157}(-)$	$0.0146_{0.0139}(-)$
	DRA	$0.0871_{0.0298}(-)$	$0.0527_{0.0363}(-)$	$0.0347_{0.0345}(-)$	$0.0240_{0.0290}(-)$	$0.0159_{0.0225}(-)$
	GRA	$\mathbf{0.0439_{0.0069}}$	$\mathbf{0.0105_{0.0022}}$	$\mathbf{0.0075_{0.0022}}$	$\mathbf{0.0061_{0.0016}}$	$\mathbf{0.0055_{0.0015}}$
F5	DE	$0.0504_{0.0072}(-)$	$0.0361_{0.0078}(-)$	$0.0277_{0.0067}(-)$	$0.0237_{0.0054}(-)$	$0.0211_{0.0052}(-)$
	DRA	$0.0587_{0.0242}(-)$	$0.0379_{0.0212}(-)$	$0.0275_{0.0154}(-)$	$0.0214_{0.0114}(-)$	$0.0183_{0.0114}(-)$
	GRA	$\mathbf{0.0423_{0.0071}}$	$\mathbf{0.0254_{0.0058}}$	$\mathbf{0.0173_{0.0035}}$	$\mathbf{0.0134_{0.0035}}$	$\mathbf{0.0116_{0.0031}}$
F6	DE	$0.1547_{0.0431}(-)$	$0.0860_{0.0076}(-)$	$0.0691_{0.0110}(-)$	$0.0597_{0.0146}(-)$	$0.0516_{0.0175}(-)$
	DRA	$0.1326_{0.0589}(\sim)$	$\mathbf{0.0742_{0.0137}}(\sim)$	$\mathbf{0.0555_{0.0190}}(\sim)$	$\mathbf{0.0383_{0.0244}}(\sim)$	$\mathbf{0.0229_{0.0268}}(\sim)$
	GRA	$\mathbf{0.1227_{0.0154}}$	$0.0750_{0.0075}$	$0.0582_{0.0205}$	$0.0433_{0.0269}$	$0.0302_{0.0304}$
F7	DE	$0.7619_{0.1405}(-)$	$0.6102_{0.1316}(-)$	$0.4782_{0.1887}(-)$	$0.3538_{0.2261}(-)$	$0.2794_{0.2248}(-)$
	DRA	$0.7641_{0.0888}(-)$	$0.6731_{0.0776}(-)$	$0.6115_{0.0751}(-)$	$0.5496_{0.0845}(-)$	$0.5001_{0.1027}(-)$
	GRA	$\mathbf{0.6405_{0.1348}}$	$\mathbf{0.3797_{0.1340}}$	$\mathbf{0.1444_{0.1092}}$	$\mathbf{0.0438_{0.0466}}$	$\mathbf{0.0138_{0.0163}}$
F8	DE	$0.2567_{0.0502}(-)$	$0.1074_{0.0514}(-)$	$0.0650_{0.0436}(-)$	$0.0435_{0.0364}(-)$	$0.0315_{0.0311}(-)$
	DRA	$0.2432_{0.0811}(-)$	$0.1482_{0.0784}(-)$	$0.1009_{0.0667}(-)$	$0.0722_{0.0578}(-)$	$0.0537_{0.0492}(-)$
	GRA	$\mathbf{0.1820_{0.0414}}$	$\mathbf{0.0656_{0.0398}}$	$\mathbf{0.0279_{0.0179}}$	$\mathbf{0.0125_{0.0065}}$	$\mathbf{0.0079_{0.0038}}$
F9	DE	$0.1669_{0.0778}(-)$	$0.0849_{0.0572}(-)$	$0.0342_{0.0236}(-)$	$0.0152_{0.0053}(-)$	$0.0102_{0.0022}(-)$
	DRA	$0.1812_{0.0698}(-)$	$0.0556_{0.0271}(-)$	$0.0168_{0.0069}(-)$	$0.0099_{0.0018}(-)$	$0.0078_{0.0010}(-)$
	GRA	$\mathbf{0.0804_{0.0419}}$	$\mathbf{0.0135_{0.0039}}$	$\mathbf{0.0080_{0.0012}}$	$\mathbf{0.0062_{0.0007}}$	$\mathbf{0.0053_{0.0006}}$
UF1	DE	$0.0497_{0.0198}(-)$	$0.0088_{0.0025}(-)$	$0.0048_{0.0009}(-)$	$0.0038_{0.0007}(-)$	$0.0032_{0.0005}(-)$
	DRA	$0.0543_{0.0386}(-)$	$0.0117_{0.0092}(-)$	$0.0062_{0.0013}(-)$	$0.0048_{0.0009}(-)$	$0.0040_{0.0007}(-)$
	GRA	$\mathbf{0.0246_{0.0086}}$	$\mathbf{0.0047_{0.0006}}$	$\mathbf{0.0031_{0.0004}}$	$\mathbf{0.0025_{0.0003}}$	$\mathbf{0.0022_{0.0002}}$
UF2	DE	$0.0337_{0.0062}(-)$	$0.0221_{0.0060}(-)$	$0.0160_{0.0047}(-)$	$0.0124_{0.0039}(-)$	$0.0105_{0.0037}(-)$
	DRA	$0.0391_{0.0220}(\sim)$	$0.0213_{0.0079}(-)$	$0.0145_{0.0060}(-)$	$0.0105_{0.0040}(-)$	$0.0084_{0.0035}(-)$
	GRA	$\mathbf{0.0300_{0.0049}}$	$\mathbf{0.0116_{0.0024}}$	$\mathbf{0.0060_{0.0017}}$	$\mathbf{0.0049_{0.0018}}$	$\mathbf{0.0042_{0.0014}}$
UF3	DE	$0.1629_{0.0336}(-)$	$0.0540_{0.0275}(-)$	$0.0230_{0.0181}(-)$	$0.0121_{0.0116}(-)$	$0.0065_{0.0040}(-)$
	DRA	$0.1462_{0.0662}(\sim)$	$0.0744_{0.0531}(-)$	$0.0368_{0.0350}(-)$	$0.0169_{0.0225}(-)$	$0.0092_{0.0134}(-)$
	GRA	$\mathbf{0.1320_{0.0497}}$	$\mathbf{0.0389_{0.0311}}$	$\mathbf{0.0119_{0.0084}}$	$\mathbf{0.0044_{0.0022}}$	$\mathbf{0.0028_{0.0010}}$
UF4	DE	$0.1382_{0.0082}(-)$	$0.1222_{0.0075}(-)$	$0.1144_{0.0074}(-)$	$0.1091_{0.0072}(-)$	$0.1056_{0.0070}(-)$
	DRA	$0.1405_{0.0109}(-)$	$0.1272_{0.0101}(-)$	$0.1192_{0.0089}(-)$	$0.1142_{0.0078}(-)$	$0.1102_{0.0072}(-)$
	GRA	$\mathbf{0.1123_{0.0087}}$	$\mathbf{0.1006_{0.0066}}$	$\mathbf{0.0957_{0.0060}}$	$\mathbf{0.0924_{0.0054}}$	$\mathbf{0.0901_{0.0050}}$
UF5	DE	$0.8950_{0.0344}(\sim)$	$0.8032_{0.0799}(-)$	$0.6860_{0.0832}(-)$	$0.5992_{0.0781}(-)$	$0.5566_{0.0761}(-)$
	DRA	$\mathbf{0.8650_{0.0627}}(+)$	$\mathbf{0.7041_{0.0872}}(\sim)$	$0.5948_{0.0782}(-)$	$0.5335_{0.0927}(-)$	$0.4954_{0.0926}(\sim)$
	GRA	$0.8957_{0.0354}$	$0.7061_{0.0796}$	$\mathbf{0.5385_{0.0644}}$	$\mathbf{0.4837_{0.0539}}$	$\mathbf{0.4684_{0.0626}}$
UF6	DE	$0.4872_{0.0665}(+)$	$0.2970_{0.1044}(\sim)$	$0.2830_{0.1071}(\sim)$	$0.2789_{0.1073}(\sim)$	$0.2759_{0.1071}(\sim)$
	DRA	$\mathbf{0.4017_{0.1345}}(+)$	$0.3825_{0.1444}(-)$	$0.3801_{0.1451}(-)$	$0.3774_{0.1442}(-)$	$0.3703_{0.1442}(-)$
	GRA	$0.5229_{0.0857}$	$\mathbf{0.2782_{0.0946}}$	$\mathbf{0.2658_{0.0927}}$	$\mathbf{0.2561_{0.0919}}$	$\mathbf{0.2443_{0.0906}}$
UF7	DE	$0.0435_{0.0181}(-)$	$0.0176_{0.0060}(-)$	$0.0095_{0.0031}(-)$	$0.0065_{0.0019}(-)$	$0.0050_{0.0012}(-)$
	DRA	$0.0386_{0.0471}(-)$	$0.0144_{0.0104}(-)$	$0.0098_{0.0099}(-)$	$0.0079_{0.0099}(-)$	$0.0069_{0.0098}(-)$
	GRA	$\mathbf{0.0201_{0.0050}}$	$\mathbf{0.0054_{0.0008}}$	$\mathbf{0.0036_{0.0005}}$	$\mathbf{0.0029_{0.0004}}$	$\mathbf{0.0026_{0.0003}}$
UF8	DE	$0.4043_{0.0430}(-)$	$0.1852_{0.0190}(-)$	$0.1513_{0.0082}(\sim)$	$0.1332_{0.0126}(\sim)$	$0.1154_{0.0148}(-)$
	DRA	$0.2287_{0.0752}(-)$	$0.1687_{0.0385}(-)$	$\mathbf{0.1410_{0.0207}}(\sim)$	$0.1246_{0.0245}(\sim)$	$\mathbf{0.1071_{0.0254}}(\sim)$
	GRA	$\mathbf{0.2136_{0.0170}}$	$\mathbf{0.1592_{0.0089}}$	$0.1438_{0.0188}$	$\mathbf{0.1239_{0.0268}}$	$0.1104_{0.0312}$
UF9	DE	$0.5334_{0.1013}(-)$	$0.2101_{0.0550}(-)$	$0.1234_{0.0419}(-)$	$0.0884_{0.0339}(-)$	$0.0654_{0.0189}(-)$
	DRA	$0.3187_{0.0630}(-)$	$0.2651_{0.0792}(-)$	$0.2225_{0.0858}(-)$	$0.1969_{0.0977}(-)$	$0.1819_{0.1053}(-)$
	GRA	$\mathbf{0.1864_{0.0573}}$	$\mathbf{0.0844_{0.0507}}$	$\mathbf{0.0712_{0.0529}}$	$\mathbf{0.0629_{0.0445}}$	$\mathbf{0.0596_{0.0452}}$
UF10	DE	$1.1976_{0.0000}(\sim)$	$1.1976_{0.0000}(\sim)$	$1.1976_{0.0000}(\sim)$	$1.1944_{0.0069}(+)$	$1.1820_{0.0208}(+)$
	DRA	$\mathbf{1.1790_{0.0283}}(+)$	$\mathbf{1.1050_{0.0611}}(+)$	$\mathbf{1.0302_{0.0609}}(+)$	$\mathbf{0.9703_{0.0479}}(+)$	$\mathbf{0.9325_{0.0514}}(+)$
	GRA	$1.1976_{0.0000}$	$1.1976_{0.0000}$	$1.1975_{0.0006}$	$1.1952_{0.0099}$	$1.1850_{0.0331}$

where $||\cdot||^2$ denotes the Euclidean distance, and

$$I_H^-(P, P^*, F^*) = I_H(P^*, F^*) - I_H(P, F^*)$$

where $I_H(P, F^*)$ calculates the hypervolume between P and F^*, i.e., the volume covered by P and F^*. Both IGD and I_H^- metrics can measure the diversity and convergence of the obtained approximations. To have low IGD and I_H^- values, the obtained approximations should be as close to the PF as possible and as diverse as possible. In the experiments, about 100 000 evenly distributed points are selected from the PF to be the reference PF P^*, and (1.2, 1.2) and (1.2, 1.2, 1.2) are the nadir points for bi-objective and tri-objective problems, respectively.

The parameters are as follows.

1) *MOEA/D-DE:* The population size $N = 300$ for bi-objective problems and $N = 630$ for tri-objective problems, the update size $n_r = 2$, the neighborhood size $T = 30$, and the neighborhood search probability $\delta = 0.9$. The parameters are the same as in [17].
2) *MOEA/D-DRA:* The population size $N = 300$ for bi-objective problems and $N = 630$ for tri-objective problems, the update size $n_r = 0.01N$, the neighborhood size $T = 0.1N$, and the neighborhood search probability

TABLE V
WILCOXON TEST RESULTS FOR THE THREE MOEA/D VARIANTS AFTER 20%, 40%, 60%, 80%, AND 100% FEs OVER 51 RUNS

	Wilcoxon	IGD					I_H^-				
	Test	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
DE	$\sim$	3	0	1	1	2	2	2	3	2	1
	$+$	2	3	3	3	1	1	0	0	1	1
	$-$	14	16	15	15	16	16	17	16	16	17
DRA	$\sim$	3	0	0	1	1	3	2	2	2	3
	$+$	3	3	3	2	2	3	1	1	1	1
	$-$	13	16	16	16	16	13	16	16	16	15

TABLE VI
NUMBER OF CORRESPONDING RANKS FOR THE THREE MOEA/D VARIANTS AFTER 20%, 40%, 60%, 80%, AND 100% FEs OVER 51 RUNS

	Rank	IGD					I_H^-				
		20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
DE	1	0	2	2	1	1	0	0	0	0	0
	2	9	12	10	13	13	11	11	10	10	11
	3	10	5	7	5	5	8	8	9	9	8
	mean	2.53	2.16	2.26	2.21	2.21	2.42	2.42	2.47	2.47	2.42
DRA	1	3	1	1	2	2	3	3	3	2	3
	2	9	7	7	5	5	7	5	5	7	5
	3	7	11	11	12	12	9	11	11	10	11
	mean	2.21	2.53	2.53	2.53	2.53	2.32	2.42	2.42	2.42	2.42
GRA	1	16	16	16	16	16	16	16	16	17	16
	2	1	0	0	0	0	0	2	3	1	2
	3	2	3	3	3	3	3	1	0	1	1
	mean	1.26	1.32	1.32	1.32	1.32	1.32	1.21	1.16	1.16	1.21

$\delta = 0.9$. The last two parameters are the same as in [35].

3) *MOEA/D-GRA:* The population size $N = 300$ for bi-objective problems and $N = 630$ for tri-objective problems, the neighborhood size $T = 20$, the neighborhood search probability $\delta = 0.8$, and the history length $\Delta T = 20$.
4) The number of FEs are 1.5×10^5 for F1-F5 and F7-F9, and 3.0×10^5 for F6 and UF1-UF10.
5) The number of executions is 51.

To save space, we use DE, DRA, and GRA to denote MOEA/D-DE, MOEA/D-DRA, and MOEA/D-GRA, respectively, in Tables III–VI.

A. Experimental Results

Tables III and IV show the mean and standard deviation of the IGD and I_H^- metric values, respectively. The values are recorded after 20%, 40%, 60%, 80%, and 100% FEs.

Table III shows that: 1) on F6 and UF10, MOEA/D-DRA can achieve the best results; 2) on UF8, although MOEA/D-DE achieves the best mean value, the Wilcoxon test indicates there is no significant difference between the three algorithms; and 3) on all the other sixteen problems, MOEA/D-GRA outperforms both MOEA/D-DE and MOEA/D-DRA.

Table IV presents the statistical results on the I_H^- metric. Table IV shows that: 1) on F6, UF8, and UF10, MOEA/D-DRA can achieve the best results, however, the Wilcoxon test shows that on F6 and UF8, MOEA/D-DRA and MOEA/D-GRA perform similarly, and 2) on all the other 16 problems, MOEA/D-GRA outperforms both MOEA/D-DE and MOEA/D-DRA.

From the two tables, we can also see that MOEA/D-GRA works very stably. If MOEA/D-GRA wins finally, it will win in the very early stages in most cases. The reason might be the ONRA strategy with the relative improvement utility function can successfully monitor the search behavior and therefore adjust the computational resources.

B. Statistical Analysis

Based on the results in Tables III and IV, we also do some statistical analysis.

Firstly, we apply the Wilcoxon nonparametric statistical test on the algorithms and the results are shown in Table V. MOEA/D-DE and MOEA/D-DRA are compared with MOEA/D-GRA, respectively. In the table, $\sim$, $+$, and $-$ denote that the algorithm is equal, better, or worse than MOEA/D-GRA with a 5% significance level. Comparing MOEA/D-DE and MOEA/D-GRA, we can see that according to the IGD metric, MOEA/D-DE performs similarly to or outperforms MOEA/D-GRA on at most five problems in all the running stages; according to the I_H^- metric, MOEA/D-DE performs similarly to or better than MOEA/D-GRA on at most three problems in all the running stages. This suggests that MOEA/D-GRA can significantly improve the performance of MOEA/D-DE. Comparing MOEA/D-DRA and MOEA/D-GRA, we can obtain the same conclusion. This indicates that the GRA strategy proposed in this paper works better than the dynamic resource allocation strategy proposed in [35]. Moreover, the proposed strategy has less control parameters than the dynamic resource allocation strategy. It should also be noted that with the given computational resources, the three algorithms can tackle most of the 19 test problems successfully. The results are reasonable because basically the three algorithms share the same framework, given enough computational resource, they should achieve similar final results.

Secondly, we analyze the results from another aspect by comparing the ranks of the three algorithms. Table VI shows the statistical results. The mean value denotes the mean rank value over the 19 problems. For MOEA/D-DE, the number of rank 1 changes from 2 (40% FE) to 1 (100% FE) according to the IGD metric, and there is no rank 1 run according to the I_H^- metric; while the number of rank 3 changes from 10 (20% FE) to 5 (100% FE) according to the IGD metric, and is around 8 according to the I_H^- metric. For MOEA/D-DRA, the number of rank 1 is between 1 to 3 according to the IGD metric, and is 2 or 3 according to the I_H^- metric; while the number of rank 3 is between 7 and 12 according to the IGD metric, and is between 9 to 11 according to the I_H^- metric. For MOEA/D-GRA, the number of rank 1 is more than 16 according to both IGD and I_H^- metrics in all the running stages, and the number of rank 1 is higher than those of MOEA/D-DE and MOEA/D-DRA according to the two metric values. On average, MOEA/D-GRA achieves the smallest mean rank values and MOEA/D-DE and MOEA/D-DRA perform similarly.

From both Wilcoxon nonparametric test and rank values, we can conclude that MOEA/D-GRA performs better than both MOEA/D-DE and MOEA/D-DRA although it can not always achieve the best result on every test problem at each search stage.

VII. Conclusion

In this paper, we have proposed a GRA strategy for MOEA/D, and the resultant algorithm is called MOEA/D-GRA. A PoI vector is applied to choose subproblems to invest. The PoI vector also determines the computational resource assigned to the subproblems. This paper has focused on how to maintain the PoI vector. We have presented two resource allocation strategies under MOEA/D-GRA that are based on offline or online measurement of subproblem hardness. The experimental results have shown that the online strategy is more practical than the offline one. We have also proposed some other utility functions to monitor the algorithm search behavior and thus to allocate the computational resources. The experimental results have indicated that different strategies can be easily incorporated into the MOEA/D-GRA framework. Finally, we have compared the new approach with the original MOEA/D and a variant with dynamic resource allocation strategy and the statistical results suggested our method performs the best on most test problems.

There are some other MOEA/D variants. It is interesting to apply the GRA strategy to them. How to define more efficient and effective utility functions for different problems is also worth further investigation.

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.

[2] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[3] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

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

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Prob.*, Athens, Greece, 2002, pp. 95–100.

[6] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

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

[8] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, Jun. 2003.

[9] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

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

[11] M. Basseur and E. Zitzler, "Handling uncertainty in indicator-based multiobjective optimization," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 255–272, 2006.

[12] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.

[13] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[14] E. Zitzler, L. Thiele, and J. Bader, "On set-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 58–79, Feb. 2010.

[15] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Zurich, Switzerland, 2001, pp. 82–95.

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

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

[18] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[19] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, 2011.

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

[21] M. A. Jan and R. A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 128–148, 2012.

[22] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[23] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for multiobjective evolutionary algorithm based decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.

[24] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Vancouver, BC, Canada, 2014, pp. 645–652.

[25] H. Ishibuchi and Y. Nojima, "Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning," *Soft Comput.*, vol. 15, no. 12, pp. 2415–2434, 2010.

[26] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 151–165, Apr. 2011.

[27] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization," *Evol. Comput.*, vol. 20, no. 3, pp. 423–452, 2012.

[28] A. Konstantinidis and K. Yang, "Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4117–4134, 2012.

[29] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Physica A Statist. Mech. Appl.*, vol. 391, no. 15, pp. 4050–4060, 2012.
[30] R. Carvalho, R. R. Saldanha, B. N. Gomes, A. C. Lisboa, and A. X. Martins, "A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi–Uda antennas," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 803–806, Feb. 2012.
[31] B. Liu, Q. Zhang, F. Fernandez, and G. Gielen, "An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 786–796, Dec. 2013.
[32] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, Feb. 2014.
[33] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 682–691, Sep. 2012.
[34] M. A. Medina, S. Das, C. A. C. Coello, and J. M. Ramirez, "Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow—A comparative study," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 10–20, Jun. 2014.
[35] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Trondheim, Norway, 2009, pp. 203–208.
[36] W. Khan and Q. Zhang, "MOEA/D-DRA with two crossover operators," in *Proc. U.K. Workshop Comput. Intell. (UKCI)*, Colchester, U.K., 2010, pp. 1–6.
[37] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 1473–1480.
[38] R. Hinterding, Z. Michalewicz, and A. Eiben, "Adaptation in evolutionary computation: A survey," in *Proc. IEEE Int. Conf. Evol. Comput.*, Indianapolis, IN, USA, 1997, pp. 65–69.
[39] K. De Jong, "Parameter setting in EAs: A 30 year perspective," in *Parameter Setting in Evolutionary Algorithms* (Studies in Computational Intelligence), vol. 54. Berlin, Germany: Springer, 2007, pp. 1–18.
[40] S. Meyer-Nieberg and H.-G. Beyer, "Self-adaptation in evolutionary algorithms," in *Parameter Setting in Evolutionary Algorithms* (Studies in Computational Intelligence), vol. 54. Berlin, Germany: Springer, 2007, pp. 47–76.
[41] O. Kramer, "Evolutionary self-adaptation: A survey of operators and strategy parameters," *Evol. Intell.*, vol. 3, no. 2, pp. 51–65, 2010.
[42] J. Zhang *et al.*, "A survey on algorithm adaptation in evolutionary computation," *Front. Electr. Electron. Eng.*, vol. 7, no. 1, pp. 16–31, 2012.
[43] I. Giagkiozis, R. Purshouse, and P. Fleming, "Generalized decomposition," in *Evolutionary Multi-Criterion Optimization* (LNCS 7811), R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, Eds. Berlin, Germany: Springer, 2013, pp. 428–442.
[44] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, Oct. 2014.
[45] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[46] K. Li, Q. Zhang, S. Kwong, and M. Li, "Stable matching based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2014.
[47] A. Zhou, Q. Zhang, and G. Zhang, "Multiobjective evolutionary algorithm based on mixture Gaussian models," *J. Softw.*, vol. 25, no. 5, pp. 913–928, 2014.
[48] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
[49] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electr. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-887, 2008.
[50] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
[51] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Comput. Eng. Netw. Lab., ETH Zurich, Zurich, Switzerland, Tech. Rep. TIK 214, 2006.

Aimin Zhou (S'08–M'10) received the B.Sc. and M.Sc. degrees from Wuhan University, Wuhan, China, in 2001 and 2003, respectively, and the Ph.D. degree from University of Essex, Colchester, U.K., in 2009, all in computer science.

He is an Associate Professor with the Shanghai Key Laboratory of Multidimensional Information Processing, and the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include evolutionary computation and optimization, machine learning, image processing, and their applications. He has published over 40 peer-reviewed papers. He is an Associate Editor of the Swarm and Evolutionary Computation.

Qingfu Zhang (M'01–SM'06) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984 and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xian, China, in 1991 and 1994, respectively.

He is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, and the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., and the Changjiang Visiting Chair Professor with Xidian University, China. He is currently leading the Metaheuristic Optimization Research (MOP) Group in City University of Hong Kong. His current research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications. He has authored over 100 research publications.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. MOEA/D, a multiobjective optimization algorithm developed in his group, won the Unconstrained Multiobjective Optimization Algorithm Competition at the Congress of Evolutionary Computation 2009. He was awarded the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award for his MOEA/D paper. He is also an Editorial Board Member of three other international journals.