A Random-Based Dynamic Grouping Strategy for Large Scale Multi-objective Optimization

An Song, Qiang Yang, *Student Member, IEEE*, Wei-Neng Chen, *Member, IEEE*
and Jun Zhang, *Senior Member, IEEE*

***Abstract*—This paper presents a random-based dynamic grouping strategy (RDG) for cooperative coevolution to deal with large scale multi-objective optimization problems (MOPs) by decomposing the whole dimension into several groups of variables with an equal size. First, a decomposer pool containing different group sizes is designed. Then, a group size is dynamically selected with probability in the evolution process. The probability of each group size in the pool is computed based on the historical performance measured by C-metric, a common metric in multi-objective optimization. Under the selected group size, random grouping is executed to decompose the whole dimension into groups. Through this, both the group size and the group components are dynamic. Finally, combining RDG with a traditional and famous multi-objective evolutionary algorithm (MOEA) named MOEA/D, we develop MOEA/D-RDG to cope with large scale MOPs. The efficacy of the proposed MOEA/D-RDG is verified on two sets of MOPs (UF1-UF10 and WFG1-WFG9) through comparing with two MOEA/D variants.**

I. Introduction

Multi-objective optimization is an important and challenging field in evolutionary computation. In recent years, an increasing number of researchers have devoted their attention to solving these optimization problems and many remarkable multi-objective evolutionary algorithms (MOEAs) have emerged [1-5], such as NSGA-II [2], and MOEA/D [1].

Even though these MOEAs have achieved great success on the tested benchmarks [1, 2] and further have been applied to solve real-world problems [6, 7], they would seriously lose their efficiency in coping with large scale multi-objective problems (MOPs) [8-10]. This dilemma is caused by the exponentially increased search space when the dimensionality grows, which is called "the curse of dimensionality". However, numerous real-world applications are with high dimensionality in nature, which promotes the urgent necessity of specially designed MOEAs to cope with large scale MOPs.

To deal with high-dimensional problems efficiently, a classical approach in single-objective optimization problems is cooperative coevolution (CC) [11], which adopts a divide-and-conquer strategy to divide high-dimensional problems into low-dimensional sub-problems. After the decomposition, CC treats each sub-problem separately and optimizes them independently. When optimizing a sub-problem, the components of the other sub-problems are usually fixed as the corresponding values in the global best solution found so far, so that the function evaluation can be executed.

For CC, one key step for obtaining satisfactory performance is the decomposition strategy. Since CC considers optimizing each sub-problem separately, a good decomposition strategy should put interdependent variables into the same group and place independent variables into different groups. In our view, existing grouping strategies can be classified into two categories: 1) fixed grouping strategies [12-14] and 2) dynamic grouping strategies [15-17], which will be reviewed in detail in next section.

Fixed grouping strategies [12, 13] usually consider the dependency among variables as global information and thus take a lot of function evaluations to detect such dependency. After decomposition, the variables in each group remain unchanged during the whole optimization process. On the contrary, in dynamic grouping strategies [15-17], the dependency among variables is taken as local information, and thus the variables in each group are dynamically determined. Generally, these dynamic strategies cooperate with evolutionary optimizers in the evolution process, and thus they cost no special function evaluations.

Utilizing the idea of fixed grouping strategies, Ma *et al*. [8] proposed an MOEA/D variant named MOEA/DVA, which combines CC with MOEA/D [1] to solve large scale MOPs. Even though this algorithm is promising for large scale MOPs, it drastically loses its feasibility when the dimensionality increases up to 1000 [8]. The inefficiency is mainly caused by the huge consumption of function evaluations in the grouping process, especially when the number of objectives is large. Different from single objective optimization [18-21,30] where variable dependency exists in only one objective, in MOPs, variable dependency lies in multiple objectives, leading to the increased difficulty in the decomposition. Under such an environment, considering the variable dependency as global information is not that advisable for large scale MOPs.

Therefore, from the perspective of dynamic grouping strategies, in this paper, we propose a random-based dynamic grouping (RDG) strategy for cooperative coevolution to cope with large scale MOPs. In the proposed RDG, not only the variables in each group but also the group size are dynamically determined. First, a decomposer pool containing different group sizes is designed. Then, before evolution, a

An Song, Qiang Yang, Wei-Neng Chen, and Jun Zhang are with School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, 510006. Corresponding authors: Wei-Neng Chen (email: cwnraul634@aliyun.com) and Jun Zhang (email: junzhang@ieee.org)

An Song and Qiang Yang are also students with School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China, 510006.

This work was supported in part by the NSFC projects Nos. 61379061, 61332002, 6141101191, in part by Natural Science Foundation of Guangdong for Distinguished Young Scholars No. 2015A030306024, in part by the "Guangdong Special Support Program" No. 2014TQ01X550, and in part by the Guangzhou Pearl River New Star of Science and Technology No. 151700098.

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

group size is probabilistically chosen from the pool and then random grouping is utilized to decompose the whole dimensions into groups. Through this, RDG goes along with the evolution process and thus no special function evaluations are needed in the decomposition, which saves a lot of function evaluations for evolution. Then, embedding RDG into a famous MOEA, named MOEA/D [1], MOEA/D-RDG is developed in this paper.

The effectiveness and superiority of the proposed MOEA/D-RDG is verified by comparing with MOEA/D [1] and MOEA/DVA [8], a state-of-the-art MOEA/D variant dealing with large scale MOPs, on two widely used MOPs benchmark sets (UF1-UF9 and WFG1-WFG9).

The rest of this paper is organized as follows. Section II reviews some related MOEAs and grouping strategies for CC. The detailed elaboration of the proposed MOEA/D-RDG is presented in Section III. In Section IV, experiments are conducted to verify the efficiency and effectiveness of MOEA/D-RDG. Finally, Section V concludes this paper.

II. Related Work

Without loss of generality, an MOP can be formulated as:

$$\begin{cases} \min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})) \\ \text{subject to: } \boldsymbol{x} \in \boldsymbol{\Omega} \end{cases} \tag{1}$$

where $\boldsymbol{F}(\boldsymbol{x})$ consists of m real-valued continuous objectives $(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x}))$, $\boldsymbol{x} \in \boldsymbol{R}^D$ is the decision vector with D dimensions and $\boldsymbol{\Omega}$ is the domain of decision vector $\boldsymbol{x}$.

Generally, in multi-objective optimization, many optimal solutions exist, which cannot dominate each other. The dominance between two solutions ($\boldsymbol{x}$ and $\boldsymbol{y}$) can be defined as:

$$\begin{aligned} &\forall i \in (1,2,\dots,M): F_i(\boldsymbol{x}) \le F_i(\boldsymbol{y}) \wedge \\ &\exists j \in (1,2,\dots,M): F_j(\boldsymbol{x}) < F_j(\boldsymbol{y}) \end{aligned} \tag{2}$$

if the above condition is satisfied, $\boldsymbol{x}$ dominates $\boldsymbol{y}$, denoted by $\boldsymbol{x} \prec \boldsymbol{y}$. A decision vector $\boldsymbol{x}$ that is not dominated by any other vector is called Pareto optimal. The set of the Pareto optimal solutions in the decision space is called Pareto set (denoted as PS) and the corresponding set of objective vectors is called Pareto front (denoted as PF).

In general, we cannot obtain all the Pareto optimal solutions in $\boldsymbol{\Omega}$ and all the objective vectors on PF. Thus, the aim of multi-objective optimization lies in two aspects: 1) the obtained PF should approximate to the true PF as much as possible; and 2) the obtained objective vectors should be distributed on the PF as diversely as possible.

A. Multi-objective Evolutionary Algorithms

In recent years, multi-objective optimization has drawn much attention and many remarkable MOEAs [1, 3, 10, 22] have emerged. Generally speaking, MOEAs can be categorized into two classes: 1) dominance-based MOEAs [2, 3]; and 2) decomposition-based MOEAs [1, 8].

As for the dominance-based MOEAs, the most popular one is NSGA-II [2]. The algorithm uses a non-dominated sorting approach along with an elitist learning strategy and a crowding mechanism to preserve the diversity of the population. After generating *NP* (*NP* is the population size) offspring using GA operators, the parents and the offspring are combined together and the above three techniques are adopted to select *NP* individuals to enter the next generation. Following this algorithm, many variants [3, 5, 23, 24] were proposed, such as NSGA-III [3], and θ-DEA [5].

In terms of decomposition-based MOEAs, the most representative one is MOEA/D [1]. Its main idea is to transform multi-objective optimization into single-objective optimization by putting weights on each objective. Utilizing the Tchebycheff approach, MOEA/D transforms an MOP into *NP* single-objective problems. Then, *NP* individuals are generated and evolved cooperatively with each individual responsible for one problem. Subsequently, many researchers have developed other new decomposition-based MOEAs, such as MOEA/D-DI [22] and DMOGA-OBL [25].

Even though the above MOEAs are promising for MOPs, they all are effective just in low dimensional space (less than 100 dimensions). When it comes to high- dimensional MOPs (more than 500 dimensions), they would drastically lose their performance [8,9]. This is mainly caused by the exponentially increased search space when the dimensionality grows. On one hand, the wide search space greatly promotes the necessity of a diverse population, so that individuals can explore all the promising space; On the other hand, usually many local PFs exist in the high-dimensional space, leading to easy stagnation or trapping into local areas. Both situations challenge the existing MOEAs. Therefore, to deal with large scale MOPs efficiently, special MOEAs should be designed.

B. Grouping Strategies for Cooperative Coevolution

In this part, we review some grouping strategies for cooperative coevolution, which have been frequently used to deal with large scale single objective optimization so far.

Cooperative coevolution (CC) [11] adopts a *divide-and-conquer* strategy to decompose high-dimensional problems into several low-dimensional sub-problems. Then, it treats each sub-problem separately and optimizes them individually.

Due to this mechanism, one key step for CC is the grouping strategy, which divides dimensions into groups. In general, a good grouping strategy should put interdependent variables into the same group and meanwhile place independent variables into different groups [12]. So far, to the best of our knowledge, existing grouping strategies can be classified into two categories: 1) fixed grouping strategies [12-14] and 2) dynamic grouping strategies [15-17].

As for fixed grouping strategies, the most popular one is differential grouping (DG) [12], which adopts Eq. (3) to detect variable dependency between two variables and then divides the dimensions into groups based on the detected dependency.

In DG, x_i and x_j are interdependent if the below inequation is satisfied:

$$\begin{aligned} &f(x_1,,,x_i+\delta_i,,,x_j,,,x_n) - f(x_1,,,x_i,,,x_j,,,x_n) \neq \\ &f(x_1,,,x_i+\delta_i,,,x_j+\delta_j,,,x_n) - \\ &f(x_1,,,x_i,,,x_j+\delta_j,,,x_n) \end{aligned} \tag{3}$$

where $\delta_i, \delta_j \in \boldsymbol{R}$ and $\delta_i, \delta_j \neq 0$. Originally, DG can only identify decision variables that interact directly. Then, Sun *et al*.[14] developed an extension of DG, named XDG, which improves

the accuracy of grouping by identifying decision variables that also interact indirectly. Another fixed grouping strategy is the variable interaction learning strategy (VIL) [13], which uses a small population to detect variable dependency. Once these fixed grouping strategies divide the dimensions into groups according to variable dependency, the components of each group are fixed during the evolution and each group is considered as a separated sub-problem to optimize.

Apparently, these fixed grouping strategies [12-14] consider variable dependency as global information. Thus for these methods, the key is to detect the variable dependency as accurately as possible. However, usually, it takes a lot of function evaluations to detect such dependency, resulting in the reduced number of function evaluations used for evolution.

In dynamic grouping strategies [15-17], on the contrary, the variable dependency is taken as local information, and thus the variables in each group are dynamically determined. The representative one is random grouping (RG) [17], which randomly divides the whole dimensions into groups with an equal size. Following this idea, Yang *et al.* [15] proposed multilevel cooperative coevolution (MLCC) to further improve the performance of CC by letting the group size dynamically determined according to the historical information. Subsequently, Delta grouping [16] was proposed, which computes the absolute difference (delta value) of each dimension in two consecutive generations and sorts the dimensions according to their corresponding delta values. Compared with random grouping, this grouping method utilizes the heuristic information, which can reflect the variable dependency, to divide the dimensions into groups.

Generally, these dynamic strategies cooperate with evolutionary optimizers in the optimization process and thus no special function evaluations are needed in the grouping process, which is beneficial for the evolution process.

Utilizing the idea of the fixed grouping strategies, Ma *et al.* [8] proposed a decision variables analysis method (DVA) for CC to deal with large scale MOPs. Combining DVA with MOEA/D, they developed MOEA/DVA and showed the superiority of MOEA/DVA to MOEA/D. In MOEA/DVA, three stages exist. The first is the decision variable analysis, where variables are divided into converged variables and diverse variables and then converged variables is divided into groups according to variable dependency. The second stage is cooperative coevolution, where each sub-group is evolved individually. The last stage is the uniform optimization, where all variables including converged and diverse variables are evolved uniformly using MOEA/D.

Even though MOEA/DVA is promising for large scale MOPs, it drastically loses its performance when the dimensionality increases to 1000. The inefficiency is mainly caused by the huge consumption of function evaluations in decomposition process. Specifically, MOEA/DVA needs $1/2\times D(D\text{-}1)\times 3\times NIA\times m$ (m is the number of objectives, NIA is the maximum number of tries required to judge the dependency between two variables and D is the dimension size) function evaluations to divide dimensions into groups.

In addition, different from single objective optimization, where the variable dependency exists in only one objective, in large scale MOPs, the variable dependency lies in multiple objectives, which greatly increases the difficulty in decomposition. Under such an environment, considering the variable dependency as global information is not that advisable for large scale MOPs.

Therefore, from view point of dynamic grouping strategies, in this paper, we propose a random-based dynamic grouping strategy for CC to deal with large scale MOPs, which will be detailed in next section.

III. Random-based Dynamic Grouping Strategy

As aforementioned, in MOPs, variable dependency lies in many objectives, which greatly enhances the difficulty in grouping dimensions accurately. In addition, the variable dependency in one objective may not be suitable for other objectives. Furthermore, in single objective optimization, detecting variable dependency accurately usually costs a large number of function evaluations [12, 14]. Such situation is much more serious in MOPs. Thus, considering the variable dependency as global information as in fixed grouping strategies [8] for CC, is not suitable for large scale MOPs.

As a consequence, as far as we are concerned, viewing variable dependency as local information is preferred in CC for large scale MOPs. In this paper, enlightened by the dynamic grouping strategies [15, 17] in single objective optimization, we propose a simple random-based dynamic grouping (RDG) strategy for CC to deal with large scale MOPs. In RDG, not only the variables in each group are dynamically determined, but also the group size is dynamically selected from a group size pool.

A. RDG

Taking inspiration from a dynamic grouping strategy named MLCC [15] in single objective optimization, we first predefine a group size pool $\boldsymbol{S}=\{s_1,s_2,\ldots,s_k\}$ containing k different group sizes. Then, before each generation, we will select a group size from the pool with probability. To make a proper choice, we utilize the performance improvement that is caused by the usage of one group size to determine its probability, which in turn influences the selection of this group size in the next generation.

To compute the probability of each group size, we define a performance improvement list $\boldsymbol{R}=\{r_1,r_2,\ldots,r_k\}$ with the ith component corresponding to the ith group size in $\boldsymbol{S}$. This list $\boldsymbol{R}$ is used to record the relative performance improvement caused by each group size. Initially, each element of $\boldsymbol{R}$ is set to 1, indicating that each group size has the equal probability to choose at the initial stage. Then, during the evolution, the relative performance improvement of the used group size (suppose it be s_i) can be computed as:

$$r_i = C(\boldsymbol{A},\boldsymbol{B}) = \frac{|\{\boldsymbol{u}\in\boldsymbol{B}\mid\exists\boldsymbol{v}\in\boldsymbol{A}:\boldsymbol{v}\prec\boldsymbol{u}\}|}{NP} \tag{4}$$

where NP is the population size, $\boldsymbol{A}$ and $\boldsymbol{B}$ are two approximations to the PF of an MOP.

Here, *C-metric*, a metric in multi-objective optimization introduced in [1], is utilized to calculate the relative performance improvement. It is defined as the percentage of

the solutions in $\boldsymbol{B}$ that are dominated by at least one solution in $\boldsymbol{A}$ ($\boldsymbol{A}$ and $\boldsymbol{B}$ are with the same size). Obviously, the value of $C(\boldsymbol{A},\boldsymbol{B})$ is within [0,1] and $C(\boldsymbol{A},\boldsymbol{B})=1$ means that all solutions in $\boldsymbol{B}$ are dominated by some solutions in $\boldsymbol{A}$ and $C(\boldsymbol{A},\boldsymbol{B})=0$ indicates that no solution in $\boldsymbol{B}$ is dominated by a solution in $\boldsymbol{A}$.

On the basis of the calculated performance improvement $\boldsymbol{R}$, the probability of each group size $\boldsymbol{P}=\{p_1,p_2,,p_k\}$ can be computed as in [15]:

$$p_i = \frac{e^{7*r_i}}{\sum_{j=1}^{k} e^{7*r_j}}, (i = 1, 2, , , k) \tag{5}$$

Subsequently, based on the calculated probability, the roulette wheel selection method is utilized to select a group size from $\boldsymbol{S}$. Then, random grouping [17] is used to decompose the dimensions into groups.

Algorithm 1: RDG

Input: The group size pool: $\boldsymbol{S}=\{s_1,s_2,\ldots,s_k\}$; the relative performance improvement: $\boldsymbol{R}=\{r_1,r_2,\ldots,r_k\}$.

1: Compute the probability $\boldsymbol{P}$ of each group size according to Eq. (5);
2: Use the roulette wheel selection method to select a group size (gs) from $\boldsymbol{S}$;
3: Use random grouping to decompose the whole dimensions into groups $\boldsymbol{G}$

Output: Group size gs and the groups $\boldsymbol{G}$

Overall, the complete procedure of the proposed RDG is presented in **Algorithm 1**.

B. Combination with MOEA/D

For simplicity, we also embed the proposed RDG into MOEA/D [1] as in MOEA/DVA [8], so that fair comparison between these two algorithms can be obtained. Thus, a new MOEA/D variant named MOEA/D-RDG is developed, whose complete framework is presented in **Algorithm 2**.

Algorithm 2: MOEA/D-RDG

Input: The maximal number of function evaluations: *max_FE*; population size: *NP*; the threshold to terminate CC: *h*; group size pool: $\boldsymbol{S}$;

1: *FE*=0;
// first stage: decision variable analysis
2: Divided variables into diverse variables and converged variables;
3: Initialize diverse variables with uniformly random values;
4: Initialize the population by initializing the converged variables;
// second stage: cooperative coevolution
5: *u*=1;
6: **while** ($u >= h \wedge FE < max\_FE$)
7: Use **Algorithm 1** to divide the converged variables into groups;
8: **for** each group in $\boldsymbol{G}$
9: Optimize this group using CC;
10: **end for**
11: Update $\boldsymbol{R}$ according to Eq. (4);
12: Update *u* according to Eq. (6);
13: **end while**
// third stage: uniform optimization
14: **while** (*FE*<*max_FE*)
15: Evolve the population with MOEA/D uniformly (including diverse variables);
16: **end while**

Output: Objective vectors of the population

Similar to MOEA/DVA [8], the whole procedure of MOEA/D-RDG can be divided into three stages: 1) decision variables analysis; 2) cooperative coevolution; and 3) uniform optimization.

The first stage (Lines 2 to 5) is the decision variables analysis, where variables are divided into converged variables and diverse variables by analyzing variables' control properties. This stage is the same as that in MOEA/DVA [8] and can be viewed as the preparation for cooperative evolution.

The second (Lines 6 to 13) is cooperative coevolution. In this stage, converged variables are dynamically divided into different groups using RDG and then each group are evolved individually. The procedure of CC continues to optimize the converged variables in this stage until the measurement defined in Eq. (6) is below a predefined threshold or the maximum function evaluations are exhausted.

$$u = \sum_{i=1}^{NP}\sum_{j=1}^{m} \frac{|old(i,j) - cur(i,j)|}{NP} \tag{6}$$

where *NP* and *m* are the population size and the number of objectives, respectively; $old(i,j)$ and $cur(i,j)$ are the jth objective values of the ith individual in the last generation and current generation, respectively. This equation measures the accumulated difference of objective values between two consecutive generations. If this indicator is smaller than a predefined threshold (denoted as *h*), such as 0.01 adopted in both MOEA/DVA [8] and this paper, it means that no improvement can be afforded by CC.

The third stage (Lines 14 to 16) is uniform optimization. This stage can be executed only when the above indicator is smaller than the predefined threshold and the maximum function evaluations are not exhausted. In this stage, all decision variables including both converged and diverse variables are evolved uniformly using MOEA/D until the maximal number of fitness evaluations is exhausted.

Compared with MOEA/DVA, the ultimate difference between MOEA/D-RDG and MOEA/DVA is that MOEA/D-RDG adopts a dynamic grouping strategy in CC to deal with large scale MOPs. Thus, no special function evaluations are needed in grouping, leading to that more function evaluations are used for evolution compared with MOEA/DVA. This may give rise to the promising performance of MOEA/D-RDG.

TABLE I
PARAMETERS SETTINGS IN MOEA/D-RDG

D	UF1-UF7	UF8-UF10	WFG1-WFG8
800	*NP*=300	*NP*=600	*NP*=600
	$\boldsymbol{S}$={5, 10, 25, 50, 100, 200, 400}		
	max_FE=8.0E+6		
1000	*NP*=300	*NP*=600	*NP*=600
	$\boldsymbol{S}$={5, 10, 25, 50, 100, 200, 500}		
	max_FE=1.0E+7		

IV. EXPERIMENTAL STUDIES

A. Experiments setup

To verify the promising performance of MOEA/D-RDG, we conduct experiments on two widely used MOPs benchmark sets: UF1-UF10 and WFG1-WFG9. The former set contains 7 2-objective functions (UF1-UF7) and 3 3-objective functions (UF8-UF10), while the functions in the latter set are all 3-objective functions. For details of these two benchmark sets, readers are referred to [26, 27].

For fair comparison, in the experiments, we compare MOEA/D-RDG with traditional MOEA/D [1] and its variant MOEA/DVA [8], which deals with large scale MOPs.

TABLE II
COMPARISON RESULTS WITH RESPECT TO IGD ON 800-D FUNCTIONS.

No. of objectives			MOEA/D-RDG	MOEA/DVA		MOEA/D	
2	UF1	mean	4.8401E-03	**3.7674E-03**	−	1.1025E-01	+
		std	7.7090E-04	1.1931E-04		6.4854E-02	
		t-test	-	-7.4678E+00		1.2661E+01	
	UF2	mean	6.5093E-03	**2.5434E-03**	−	7.4046E-02	+
		std	2.6243E-03	1.2316E-04		3.8394E-02	
		t-test	-	-8.1978E+00		1.3639E+01	
	UF3	mean	**3.2771E-03**	1.1328E-02	+	9.6626E-02	+
		std	1.1403E-03	1.5403E-04		6.2979E-03	
		t-test	-	3.7996E+01		1.1180E+02	
	UF4	mean	1.0669E-01	**3.2929E-02**	−	6.8618E-02	−
		std	6.8523E-03	1.8726E-04		1.9041E-03	
		t-test	-	-5.8432E+01		-2.9782E+01	
	UF5	mean	**8.9196E-02**	3.2800E-01	+	1.7736E-01	+
		std	3.3778E-02	4.7797E-03		3.0535E-02	
		t-test	-	3.8014E+01		1.2041E+01	
	UF6	mean	2.6168E-02	**2.4944E-02**	−	3.4722E-01	+
		std	8.1252E-04	1.0183E-03		1.1307E-01	
		t-test	-	-5.0989E+00		2.2120E+01	
	UF7	mean	**5.0611E-03**	3.9980E-02	+	2.3463E-01	+
		std	7.0960E-04	1.8555E-04		1.3465E-01	
		t-test	-	2.5854E+02		1.3282E+01	
3	UF8	mean	**1.3749E-01**	1.9025E-01	≈	2.2281E-01	+
		std	1.1064E-01	8.8073E-02		2.7423E-02	
		t-test	-	2.0260E+00		4.1493E+00	
	UF9	mean	**2.3423E-02**	1.0222E-01	+	3.8768E-01	+
		std	8.9158E-03	4.8500E-02		4.1556E-02	
		t-test	-	8.6777E+00		6.5297E+01	
	UF10	mean	2.4460E+00	1.6657E+00	−	**3.9263E-01**	−
		std	1.5270E-01	4.3533E-01		1.9145E-01	
		t-test	-	-9.1850E+00		-5.5164E+01	
	WFG1	mean	**9.0484E-01**	1.2711E+00	+	9.8819E-01	+
		std	6.8579E-03	5.7191E-03		2.6776E-02	
		t-test	-	2.2275E+02		2.2777E+01	
	WFG2	mean	**1.1710E-01**	4.5901E-01	+	4.7508E-01	+
		std	1.8376E-02	1.2965E-04		1.2530E-01	
		t-test	-	1.0104E+02		2.1786E+01	
	WFG3	mean	1.5477E+00	1.5474E+00	≈	**1.4559E+00**	−
		std	2.2854E-02	8.3238E-03		7.2498E-03	
		t-test	-	-7.8518E-02		-2.1425E+01	
	WFG4	mean	**8.8466E-02**	4.0696E-01	+	1.2898E-01	+
		std	8.8600E-04	2.1696E-05		3.5547E-04	
		t-test	-	1.9515E+03		2.4039E+02	
	WFG5	mean	**1.1678E-01**	2.1981E-01	+	1.5346E-01	+
		std	1.8767E-04	3.0335E-05		2.1742E-04	
		t-test	-	2.9431E+03		8.2844E+02	
	WFG6	mean	**8.8343E-02**	9.8047E-02	+	1.2792E-01	+
		std	6.0161E-04	1.5948E-03		2.0785E-04	
		t-test	-	3.0919E+01		3.4920E+02	
	WFG7	mean	**8.7107E-02**	8.9486E-02	+	1.2727E-01	+
		std	1.1655E-04	2.7137E-04		2.9747E-04	
		t-test	-	4.3744E+01		9.1824E+02	
	WFG8	mean	**1.0890E-01**	1.1036E-01	+	1.4436E-01	+
		std	2.9620E-03	1.3518E-03		2.1921E-03	
		t-test	-	2.4424E+00		5.8051E+01	
	WFG9	mean	**9.0770E-02**	9.3986E-02	+	1.2738E-01	+
		std	8.5010E-04	2.8099E-03		1.0105E-03	
		t-test	-	6.0021E+00		1.5186E+02	
	w/l/t		-	12/5/2		16/3/0	

TABLE III
COMPARISON RESULTS WITH RESPECT TO IGD ON 1000-D FUNCTIONS.

No. of objectives			MOEA/D-RDG	MOEA/DVA		MOEA/D	
2	UF1	mean	**4.6088E-03**	7.4146E-03	+	1.3608E-01	+
		std	8.7010E-04	1.3857E-04		6.7160E-02	
		t-test	-	1.7443E+01		1.0721E+01	
	UF2	mean	5.4708E-03	**4.7151E-03**	≈	8.8405E-02	+
		std	2.5971E-03	1.2915E-04		3.9920E-02	
		t-test	-	-1.5918E+00		1.1355E+01	
	UF3	mean	**2.6605E-03**	1.5868E-02	+	9.4917E-02	+
		std	6.5686E-04	1.6444E-04		4.1921E-03	
		t-test	-	1.0684E+02		1.1908E+02	
	UF4	mean	1.0796E-01	**3.5739E-02**	−	7.0967E-02	−
		std	1.2308E-02	1.7990E-04		2.2074E-03	
		t-test	-	-3.2136E+01		-1.6205E+01	
	UF5	mean	**1.0984E-01**	3.7472E-01	+	1.8779E-01	+
		std	1.3889E-01	3.9009E-03		2.8222E-02	
		t-test	-	1.0442E+01		3.0125E+00	
	UF6	mean	**2.4528E-02**	3.7936E-02	+	3.7069E-01	+
		std	9.5862E-04	1.4289E-03		9.7413E-02	
		t-test	-	4.2680E+01		1.9463E+01	
	UF7	mean	**5.3030E-03**	4.2387E-02	+	2.1554E-01	+
		std	7.2415E-04	2.2853E-04		1.3838E-01	
		t-test	-	2.6749E+02		8.3214E+00	
3	UF8	mean	**1.6902E-01**	2.2740E-01	+	2.2636E-01	+
		std	1.1688E-01	7.2874E-02		1.7307E-02	
		t-test	-	2.3216E+00		2.6582E+00	
	UF9	mean	**2.6760E-02**	1.5391E-01	+	3.6633E-01	+
		std	1.6021E-02	7.3275E-02		7.7742E-02	
		t-test	-	9.2853E+00		2.3432E+01	
	UF10	mean	2.5661E+00	2.0701E+00	−	**4.1549E-01**	−
		std	2.0235E-01	1.8933E-01		1.8551E-01	
		t-test	-	-9.8040E+00		-4.2910E+01	
	WFG1	mean	**9.0189E-01**	1.2975E+00	+	9.8979E-01	+
		std	1.2965E-02	4.2976E-03		3.3410E-02	
		t-test	-	1.5866E+02		1.3434E+01	
	WFG2	mean	**1.2485E-01**	4.8137E-01	+	4.0122E-01	+
		std	2.4283E-02	3.1773E-04		1.4049E-01	
		t-test	-	8.0410E+01		1.0617E+01	
	WFG3	mean	1.5543E+00	1.5170E+00	−	**1.4483E+00**	−
		std	1.7457E-02	6.0402E-03		8.6273E-03	
		t-test	-	-1.1068E+01		-2.9810E+01	
	WFG4	mean	**8.8967E-02**	4.1518E-01	+	1.2937E-01	+
		std	1.4063E-03	1.1038E-04		1.9534E-04	
		t-test	-	1.2666E+03		1.5587E+02	
	WFG5	mean	**1.1670E-01**	2.3881E-01	+	1.5345E-01	+
		std	2.2352E-04	9.8514E-05		4.1166E-04	
		t-test	-	2.7380E+03		4.2974E+02	
	WFG6	mean	**8.8139E-02**	1.2884E-01	+	1.2788E-01	+
		std	5.1434E-04	4.6958E-03		2.5471E-04	
		t-test	-	4.7189E+01		3.7920E+02	
	WFG7	mean	**8.7095E-02**	1.1312E-01	+	1.2733E-01	+
		std	1.1844E-04	2.6257E-03		2.5389E-04	
		t-test	-	5.4226E+01		7.8663E+02	
	WFG8	mean	**1.0856E-01**	1.2317E-01	+	1.4404E-01	+
		std	3.1555E-03	3.6351E-03		2.1181E-03	
		t-test	-	1.6630E+01		5.1139E+01	
	WFG9	mean	**9.0841E-02**	1.1299E-01	+	1.2741E-01	+
		std	1.0489E-03	4.4261E-03		9.4187E-04	
		t-test	-	2.6676E+01		1.4208E+02	
	w/l/t		-	15/3/1		16/3/0	

To evaluate the performance of each compared algorithm, two commonly used measurements: the Inverted Generational Distance (IGD) [28] and Hyper Volume (HV) [29] are utilized.

Generally, IGD can be computed as follows:

$$IGD = \frac{\sum_{v \in \boldsymbol{P}^*} d(\boldsymbol{v}, \boldsymbol{P})}{|\boldsymbol{P}^*|} \tag{7}$$

where $\boldsymbol{P}^*$ is the real PF, while $\boldsymbol{P}$ is the approximated PF; $d(\boldsymbol{v},\boldsymbol{P})$ is the minimal Euclidean distance from $\boldsymbol{v}$ in $\boldsymbol{P}^*$ to $\boldsymbol{P}$, and $|.|$ is the size of a set. IGD reflects the average distance from $\boldsymbol{P}^*$ to $\boldsymbol{P}$ and the diversity of $\boldsymbol{P}$. Generally, the smaller the value of *IGD*, the better the performance of an algorithm.

Let $\boldsymbol{y}^*=(y_1^*,y_2^*,\ldots,y_m^*)$ be a reference point in objective space which is dominated by all solutions in $\boldsymbol{P}$. HV measures the volume of the area in which any point is dominated by solutions in $\boldsymbol{P}$ but dominates $\boldsymbol{y}^*$.The higher the value of HV, the better one algorithm performs.

In the experiments, for the 2-objective functions (UF1-UF7) in the first set, the reference point is set as $\boldsymbol{y}^*=(5,5)$. For the 3-objective functions in the first set (UF8-UF10), $\boldsymbol{y}^*=(5,5,5)$ is used. While for the 3-objective functions in the other set (WFG1-WFG9), $\boldsymbol{y}^*=(7,7,7)$ is used.

In addition, to comprehensively validate the performance of MOEA/D-RDG, we conduct experiments on the two benchmark sets with different dimension sizes. In this experiment, we set the dimension size as 800 and 1000. Accordingly, the population sizes and the maximum number of function evaluations for different functions with different dimensions are set as shown in Table I. Other parameters, such as mutation rate and crossover rate are set according to [8], since MOEA/D-RDG and MOEA/DVA have the similar framework in the evolution procedure.

Additionally, it is worth mentioning that all experiments are conducted 30 runs for statistics. During the comparison between two algorithms, the two-tailed t-test at significance α=0.05 is conducted, at which the critical t-test value with 30 samples is 2.042. Based on the t-test results, in following tables, the best results are highlighted in bold and the symbols "+", "−" and "≈" indicate that MOEA/D-RDG is

significantly better than, worse than and equivalent to the compared methods. Specially, the row named "*w/l/t*" counts the number of functions where MOEA/D-RDG is better than, worse than or similar to the compared algorithms.

TABLE IV
COMPARISON RESULTS WITH RESPECT TO HV ON 800-D FUNCTIONS.

No. of objectives			MOEA/D-RDG	MOEA/DVA		MOEA/D	
2	UF1	mean	2.4625E+01	**2.4644E+01**	−	2.3266E+01	+
		std	8.6794E-03	4.4265E-03		6.5117E-01	
		t-test	-	1.0863E+01		-1.1424E+01	
	UF2	mean	2.4620E+01	**2.4651E+01**	−	2.3854E+01	+
		std	2.0461E-02	3.0316E-03		3.0370E-01	
		t-test	-	8.2558E+00		-1.3785E+01	
	UF3	mean	**2.4662E+01**	2.4508E+01	+	2.3428E+01	+
		std	1.9055E-03	9.3588E-03		3.1197E-02	
		t-test	-	-8.8195E+01		-2.1624E+02	
	UF4	mean	2.3608E+01	**2.4088E+01**	−	2.3497E+01	+
		std	8.3342E-02	4.8877E-03		8.1340E-02	
		t-test	-	3.1511E+01		-5.2150E+00	
	UF5	mean	**2.3716E+01**	2.2295E+01	+	2.2780E+01	+
		std	5.6289E-01	7.1662E-02		4.0536E-01	
		t-test	-	-1.3723E+01		-7.3911E+00	
	UF6	mean	**2.4320E+01**	2.4256E+01	+	2.1189E+01	+
		std	4.8282E-03	1.7113E-02		1.1033E+00	
		t-test	-	-1.9635E+01		-1.5544E+01	
	UF7	mean	**2.4473E+01**	2.4425E+01	+	2.1931E+01	+
		std	4.2468E-03	4.1570E-03		1.3365E+00	
		t-test	-	-4.4458E+01		-1.0420E+01	
3	UF8	mean	**1.2362E+02**	1.1941E+02	+	1.2128E+02	+
		std	8.2446E-01	1.2938E+00		8.8907E-01	
		t-test	-	-1.5041E+01		-1.0595E+01	
	UF9	mean	**1.2446E+02**	1.2069E+02	+	1.0989E+02	+
		std	1.3394E-01	1.8249E+00		1.0942E+00	
		t-test	-	-1.1300E+01		-7.2419E+01	
	UF10	mean	6.3419E+01	6.3199E+01	≈	**1.1317E+02**	−
		std	3.1101E+00	1.1189E+01		1.1062E+01	
		t-test	-	-1.0347E-01		2.3714E+01	
	WFG1	mean	**2.5615E+02**	2.1324E+02	+	2.4780E+02	+
		std	7.2117E-01	5.4039E-02		1.2454E+00	
		t-test	-	-3.2504E+02		-3.1800E+01	
	WFG2	mean	**3.3422E+02**	3.2282E+02	+	2.7538E+02	+
		std	3.7052E+00	9.1341E-02		2.4784E+01	
		t-test	-	-1.6848E+01		-1.2861E+01	
	WFG3	mean	**2.8370E+02**	2.7565E+02	+	2.7293E+02	+
		std	1.2082E+00	2.6989E+00		7.4333E-01	
		t-test	-	-1.4908E+01		-4.1553E+01	
	WFG4	mean	**3.1569E+02**	2.9992E+02	+	3.1371E+02	+
		std	4.1392E-01	1.0399E-02		1.1245E-01	
		t-test	-	-2.0862E+02		-2.5241E+01	
	WFG5	mean	**3.0930E+02**	2.8044E+02	+	3.0791E+02	+
		std	3.1346E-02	7.2433E-03		5.7088E-02	
		t-test	-	-4.9125E+03		-1.1691E+02	
	WFG6	mean	**3.1581E+02**	3.1284E+02	+	3.1424E+02	+
		std	3.9195E-01	3.6466E-01		6.0532E-02	
		t-test	-	-3.0391E+01		-2.1693E+01	
	WFG7	mean	**3.1644E+02**	3.1519E+02	+	3.1475E+02	+
		std	1.6027E-02	6.1726E-02		9.1228E-02	
		t-test	-	-1.0760E+02		-9.9708E+01	
	WFG8	mean	**3.1097E+02**	3.1058E+02	+	3.0955E+02	+
		std	5.1233E-01	2.1732E-01		4.4143E-01	
		t-test	-	-3.7741E+00		-1.1488E+01	
	WFG9	mean	**3.0759E+02**	3.0471E+02	+	3.0261E+02	+
		std	3.2661E+00	2.0408E+00		4.0980E-01	
		t-test	-	-4.0940E+00		-8.2904E+00	
	w/l/t		-	15/3/1		18/1/0	

Together, in terms of IGD, MOEA/D-RDG is much superior to both MOEA/D and MOEA/DVA no matter on 800-dimension MOPs or on 1000-dimension MOPs. Such superiority demonstrates that the PF obtained by MOEA/D-RDG is much closer to the true PF and the solutions on the obtained PF are distributed much more diversely than the other two algorithms.

TABLE V
COMPARISON RESULTS WITH RESPECT TO HV ON 1000-D FUNCTIONS.

No. of objectives			MOEA/D-RDG	MOEA/DVA		MOEA/D	
2	UF1	mean	**2.4625E+01**	2.4615E+01	+	2.2980E+01	+
		std	8.9533E-03	5.1916E-03		6.7942E-01	
		t-test	-	-5.3610E+00		-1.3263E+01	
	UF2	mean	2.4628E+01	**2.4634E+01**	≈	2.3726E+01	+
		std	1.7420E-02	2.9794E-03		2.7235E-01	
		t-test	-	1.8172E+00		-1.8110E+01	
	UF3	mean	**2.4663E+01**	2.4462E+01	+	2.3445E+01	+
		std	1.1893E-03	7.9464E-03		1.9788E-02	
		t-test	-	-1.3669E+02		-3.3653E+02	
	UF4	mean	2.3581E+01	**2.4068E+01**	−	2.3457E+01	+
		std	9.8907E-02	3.5789E-03		8.5068E-02	
		t-test	-	2.6960E+01		-5.2060E+00	
	UF5	mean	**2.3598E+01**	2.1998E+01	+	2.2707E+01	+
		std	9.8902E-01	6.7876E-02		4.3170E-01	
		t-test	-	-8.8415E+00		-4.5194E+00	
	UF6	mean	**2.4328E+01**	2.4161E+01	+	2.0880E+01	+
		std	5.8371E-03	2.0202E-02		1.1194E+00	
		t-test	-	-4.3609E+01		-1.6872E+01	
	UF7	mean	**2.4470E+01**	2.4395E+01	+	2.2012E+01	+
		std	4.2270E-03	4.2588E-03		1.3339E+00	
		t-test	-	-6.8375E+01		-1.0096E+01	
3	UF8	mean	**1.2348E+02**	1.1815E+02	+	1.2113E+02	+
		std	8.7649E-01	8.1255E-01		6.7170E-01	
		t-test	-	-2.4465E+01		-1.1672E+01	
	UF9	mean	**1.2445E+02**	1.1842E+02	+	1.0995E+02	+
		std	1.6778E-01	2.6789E+00		1.6995E+00	
		t-test	-	-1.2310E+01		-4.6502E+01	
	UF10	mean	6.1669E+01	5.1404E+01	+	**1.1424E+02**	−
		std	4.9435E+00	5.3623E+00		9.2475E+00	
		t-test	-	-7.7084E+00		2.7461E+01	
	WFG1	mean	**2.5654E+02**	2.0876E+02	+	2.4724E+02	+
		std	8.9293E-01	3.9305E-02		1.6220E+00	
		t-test	-	-2.9281E+02		-2.7506E+01	
	WFG2	mean	**3.3281E+02**	3.1630E+02	+	2.9082E+02	+
		std	4.4860E+00	1.3561E-01		2.7828E+01	
		t-test	-	-2.0152E+01		-8.1603E+00	
	WFG3	mean	**2.8378E+02**	2.7048E+02	+	2.7210E+02	+
		std	9.9904E-01	2.3774E+00		9.1743E-01	
		t-test	-	-2.8248E+01		-4.7172E+01	
	WFG4	mean	**3.1544E+02**	2.9802E+02	+	3.1355E+02	+
		std	5.7913E-01	5.1305E-02		9.4438E-02	
		t-test	-	-1.6411E+02		-1.7660E+01	
	WFG5	mean	**3.0931E+02**	2.7803E+02	+	3.0789E+02	+
		std	2.9717E-02	3.1938E-02		8.7119E-02	
		t-test	-	-3.9277E+03		-8.4545E+01	
	WFG6	mean	**3.1591E+02**	3.0813E+02	+	3.1426E+02	+
		std	3.9110E-01	6.1686E-01		6.2173E-02	
		t-test	-	-5.8354E+01		-2.2760E+01	
	WFG7	mean	**3.1644E+02**	3.1047E+02	+	3.1474E+02	+
		std	1.5530E-02	3.9474E-01		8.7909E-02	
		t-test	-	-8.2774E+01		-1.0394E+02	
	WFG8	mean	**3.1094E+02**	3.0863E+02	+	3.0961E+02	+
		std	5.5722E-01	4.9237E-01		4.3537E-01	
		t-test	-	-1.7057E+01		-1.0345E+01	
	WFG9	mean	**3.0646E+02**	3.0083E+02	+	3.0258E+02	+
		std	2.7903E+00	1.7981E+00		4.1537E-01	
		t-test	-	-9.2917E+00		-7.5292E+00	
	w/l/t		-	17/1/1		18/1/0	

B. Simulation Results

1) IGD Comparison

Table II and Table III present the comparison results of the compared algorithms with regard to IGD on the two benchmark sets with 800 and 1000 dimensions respectively.

Obviously, from these tables, we can see that on both 800 and 1000-dimension MOPs, MOEA/D-RDG is much superior to MOEA/D and MOEA/DVA. Specifically, on the 19 800-dimension MOPs, MOEA/D-RDG dominates MOEA/DVA on 12 functions and beats MOEA/D down on 16 functions. When it comes to 1000-dimension problems, MOEA/D-RDG is superior to MOEA/DVA on 15 functions and succeeds in the competitions with MOEA/D on 16 functions.

2) HV Comparison

From the perspective of HV, Table IV and Table V present the comparison results of the compared algorithms on the two benchmark sets with 800 and 1000 dimensions respectively.

Observing these two tables, we can obtain the similar conclusion. In more details, among the 19 800-dimension MOPs, MOEA/D-RDG performs much better than MOEA/DVA on 15 functions and dominates MOEA/D on 18 functions. When it reaches 1000-dimension problems, MOEA/D-RDG is significantly better than MOEA/DVA and MOEA/D on 17 and 18 functions respectively.

The verified superiority of MOEA/D-RDG over MOEA/D and MOEA/DVA further demonstrates that no matter on

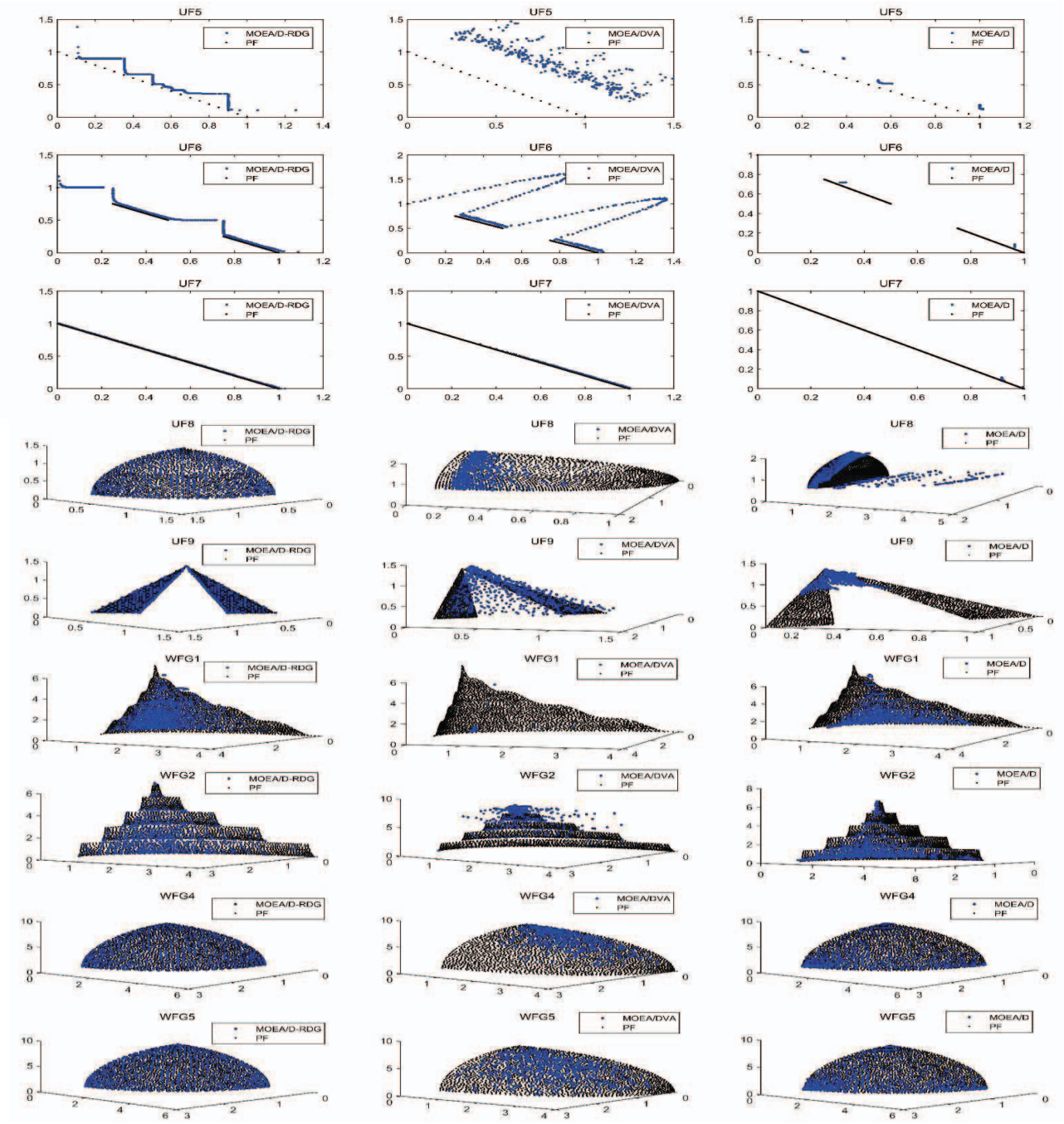

Fig. 1. Comparison results with respect to PF obtained by each algorithm on 1000-D benchmarks.

800-dimension or 1000-dimension MOPs, the PF obtained by MOEA/D-RDG is of much better quality (in terms of both the proximity to the true PF and the diversity of the solutions) than the other two methods.

3) Pareto Front Comparison

To visually understand the superiority of the proposed MOEA/D-RDG, we plot the PF obtained by each algorithm in comparison with the true PF. To save space, we only present the results of each algorithm on 9 functions (UF5-UF9, WFG1, WFG2, WFG4 and WFG5) with 1000 dimensions, which are shown in Fig. 1.

From Fig. 1, we can see that the PF obtained by MOEA/D-RDG is much better than those of the other two algorithms. On one hand, the PF of MOEA/D-RDG is much closer to the true PF than the other two algorithms. This superiority is much more obvious on functions UF5, UF9, and WFG2. On the other hand, as for the diversity of the obtained PF, we can see that the solutions on the PF of MOEA/D-RDG distribute much more diversely than the other two algorithms, especially on functions UF7, UF8, WFG1, WFG2, WFG4 and WFG5.

Overall, through decomposing the high dimensional problems into low dimensional sub-problems and evolving

them individually, MOEA/D-RDG and MOEA/DVA show their superiority to traditional MOEA/D. The superiority of MOEA/D-RDG to MOEA/DVA shows that the proposed dynamic grouping strategy, RDG, is promising in dealing with large scale MOPs.

V. Conclusions

In this paper, we have proposed a random-based dynamic grouping (RDG) strategy for cooperative coevolution to deal with large scale MOPs. Instead of taking variable dependency as global information in fixed grouping strategies, the proposed RDG considers variable dependency as local information. Not only the variables in each dimension group are dynamically determined, but also the group size is dynamically selected from a pool in RDG. Such selection is based on historical information and thus the group size can be properly determined in the evolution process.

To cope with large scale MOPs, MOEA/D-RDG is developed by embedding RDG into MOEA/D. The superiority of MOEA/D-RDG is verified by experiments conducted on two widely used MOP benchmark sets (UF1-UF10 and WFG1-WFG9) with 800 and 1000 dimensions.

This work has shown the promising of dynamic grouping strategies in dealing with MOPs, but the proposed RDG does not utilize any heuristic information or historical evolution information in decomposing dimensions into groups. Thus, it would be interesting to design dynamic grouping strategies in future work. In addition, it is also interesting to use such algorithm for real-world large-scale optimization problems such as the sensing coverage problem in wireless sensor networks [31].

References

[1] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE. Trans. Evol. Comput.,* vol. 11, pp. 712-731, 2007.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE. Trans. Evol. Comput.,* vol. 6, pp. 182-197, 2002.

[3] 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, pp. 577-601, 2014.

[4] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.,* vol. 45, pp. 2076-2088, 2014..

[5] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A New Dominance Relation Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE. Trans. Evol. Comput.,* vol. PP, pp.1, 2015.

[6] R. Shang, K. Dai, L. Jiao, and R. Stolkin, "Improved Memetic Algorithm Based on Route Distance Grouping for Multiobjective Large Scale Capacitated Arc Routing Problems," *IEEE Trans. Cybern.,* 2015.

[7] C. Liu, J. Liu and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.,* vol. 44, pp. 2274-2287, 2014.

[8] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A Multiobjective Evolutionary Algorithm based on Decision Variable Analyses for Multi-objective Optimization Problems with Large Scale Variables," *IEEE. Trans. Evol. Comput.,* vol. PP, pp.1, 2015.

[9] L. M. Antonio and C. Coello Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 2758-2765.

[10] N. Chen, W. Chen, Y. Gong, Z. Zhan, J. Zhang, Y. Li, and Y. Tan, "An evolutionary algorithm with double-level archives for multiobjective optimization," *IEEE Trans. Cybern.* vol. 45, pp. 1851-1863, 2015.

[11] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE. Trans. Evol. Comput.,* vol. 16, pp. 210-224, 2012.

[12] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE. Trans. Evol. Comput.,* vol. 18, pp. 378-393, 2014.

[13] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Parallel Problem Solving from Nature, PPSN XI*: Springer, 2010, pp. 300-309.

[14] Y. Sun, M. Kirley and S. K. Halgamuge, "Extended Differential Grouping for Large Scale Global Optimization with Direct and Indirect Variable Interactions," in *Proc. Conf. Genetic Evol. Comput.,* 2015, pp. 313-320.

[15] Z. Yang, K. Tang and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput*, 2008, pp. 1663-1670.

[16] M. N. Omidvar, X. Li and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Proc. IEEE Congr. Evol. Comput*,, pp. 1-8. 2010

[17] Z. Yang, K. Tang and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.,* vol. 178, pp. 2985-2999, 2008.

[18] W. Chen, J. Zhang, H. S. Chung, W. Zhong, W. Wu, and Y. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE. Trans. Evol. Comput.,* vol. 14, pp. 278-300, 2010.

[19] Y. Li, Z. Zhan, Y. Gong, W. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.,* vol. 45, pp. 1798-1810, 2015.

[20] W. Yu, M. Shen, W. Chen, Z. Zhan, Y. Gong, Y. Lin, O. Liu, and J. Zhang, "Differential evolution with two-level parameter adaptation," *IEEE Trans. Cybern.,* vol. 44, pp. 1080-1099, 2014.

[21] W. Chen, J. Zhang, Y. Lin, N. Chen, Z. Zhan, H. S. Chung, Y. Li, and Y. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE. Trans. Evol. Comput.,* vol. 17, pp. 241-258, 2013.

[22] Z. He and G. G. Yen, "Diversity improvement in Decomposition-Based Multi-Objective Evolutionary Algorithm for many-objective optimization problems," in *Systems, Man and Cybernetics (SMC), 2014 IEEE Int. Conf. on*, 2014, pp. 2409-2414.

[23] K. McClymont and E. Keedwell, "Deductive sort and climbing sort: New methods for non-dominated sorting," *Evol. comput.,* vol. 20, pp. 1-26, 2012.

[24] X. Zhong, W. Fan, J. Lin, and Z. Zhao, "Hybrid non-dominated sorting differential evolutionary algorithm with nelder-mead," in *WRI Global Congress on Intelligent Systems*, pp. 306-311*,* 2010.

[25] R. Patel, M. M. Raghuwanshi and L. G. Malik, "Decomposition based multi-objective genetic algorithm (DMOGA) with opposition based learning," in *International Conference on Computational Intelligence and Communication Networks* , pp. 605-610, 2012.

[26] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, technical report,*pp. 1-30, 2008.

[27] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE. Trans. Evol. Comput.,* vol. 10, pp. 477-506, 2006.

[28] Q. Zhang, A. Zhou and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE. Trans. Evol. Comput.,* vol. 12, pp. 41-63, 2008.

[29] K. Deb, *Multi-objective optimization using evolutionary algorithms* vol. 16: John Wiley & Sons, 2001.

[30] Q. Yang, W.-N. Chen, Y. Li, C. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal Estimation of Distribution Algorithms," *IEEE Trans. Cybern.*, in press, 2016.

[31] X.Y Zhang, Y.J. Gong, Z.H. Zhan, W.-N. Chen, Yun Li and J. Zhang, "Kuhn-Munkres Parallel Genetic Algorithm for the Set Cover Problem and Its Application to Large-Scale Wireless Sensor Networks", *IEEE Trans. on Evol. Comput*., in press.