An Efficient Recursive Differential Grouping for Large-Scale Continuous Problems

Ming Yang, Aimin Zhou, *Senior Member, IEEE*, Changhe Li, *Member, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*—Cooperative co-evolution (CC) is an efficient and practical evolutionary framework for solving large-scale optimization problems. The performance of CC is affected by the variable decomposition. An accurate variable decomposition can help to improve the performance of CC on solving an optimization problem. The variable grouping methods usually spend many computational resources obtaining an accurate variable decomposition. To reduce the computational cost on the decomposition, we propose an efficient recursive differential grouping (ERDG) method in this article. By exploiting the historical information on examining the interrelationship between the variables of an optimization problem, ERDG is able to avoid examining some interrelationship and spend much less computation than other recursive differential grouping methods. Our experimental results and analysis suggest that ERDG is a competitive method for decomposing large-scale continuous problems and improves the performance of CC for solving the large-scale optimization problems.**

***Index Terms*—Cooperative co-evolution (CC), decomposition, large-scale global optimization.**

I. Introduction

LARGE-SCALE optimization problems involve at least thousands of decision variables [1], [2]. It is challenging for evolutionary algorithms (EAs) [3] to solve such kind of large-scale optimization problem [4]–[6]. Cooperative co-evolution (CC) [7] adopts the divide-and-conquer strategy [8]–[10] to solve optimization problems. CC divides the variables into several subcomponents and optimizes the subcomponents separately. The divide-and-conquer strategy can decrease the difficulty of solving the large-scale optimization problems [11]–[15].

How to group the variables of a problem into subcomponents, i.e., identifying variables interaction [16], is a key problem of CC to solve. If the difference of the objective value of a problem caused by the variation of a variable is independent on the difference of the objective value caused by the variation of another variable, the two variables are separable; otherwise, the two variables are nonseparable and should be grouped together. Based on this theory, the differential grouping (DG) [17] can identify the interrelationship between a pair of variables. The experimental results in [17] showed that DG is sensitive to the value of ϵ which is a parameter of DG used for determining whether two variables are nonseparable. DG2 [18], an improved variant of DG, can adapt the value of ϵ to the objective value of a problem and improves the accuracy in identifying the interrelationship. DG and DG2 decompose a problem in a pairwise fashion at the variable level. For decomposing a D-dimensional problem, the computational complexity of DG and DG2 is $\mathcal{O}(D^2)$.

To reduce the computational cost of the DG methods, the recursive DG (RDG) [19], [20] examines the interrelationship between a pair of sets of variables but not a pair of variables. If two sets of variables (X_1 and X_2) are interrelated with each other, RDG divides X_2 into two equal-sized subsets and examines the interrelationship between X_1 and the two subsets. Repeat the above process until RDG finds the variables which interrelate with X_1. Inspired by DG2, RDG2 [21] adapts the value of ϵ to the objective value of a problem and improves the accuracy of RDG in identifying the interrelationship. The computational complexity is $\mathcal{O}(D \log_2 D)$ when RDG and RDG2 decompose a D-dimensional problem in the above binary search fashion.

In this article, we analyze the binary search process of the RDG methods and discover the association between the interrelationship examinations. To reduce the computational cost of RDG and RDG2, an improved RDG is proposed in this article. The improvements are made in the following two aspects.

1) By using the historical information on identifying the interrelationship, the improved RDG can save redundant interrelationship examinations from decomposing a problem.

Manuscript received May 29, 2019; revised January 27, 2020 and May 2, 2020; accepted July 9, 2020. Date of publication July 15, 2020; date of current version January 29, 2021. This work was supported in part by the Natural Science Foundation of Hubei Province under Grant 2019CFB584; in part by the National Natural Science Foundation of China under Grant 61305086 and Grant 61673355; and in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences, Wuhan under Grant CUGL170412. *(Corresponding author: Changhe Li.)*

Ming Yang is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, and also with the Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China (e-mail: mingyang@cug.edu.cn).

Aimin Zhou is with the 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).

Changhe Li is with the School of Automation, China University of Geosciences, Wuhan 430074, China, and also with the Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China (e-mail: lichanghe@cug.edu.cn).

Xin Yao is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: xiny@sustc.edu.cn).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TEVC.2020.3009390

2) A function evaluation (FE) is moved out from the recursive process, which does not affect the decomposition process. The improved RDG spends two FEs each time the interrelationship is examined, while RDG or RDG2 spends three FEs.

The saved computational resources can be used by an algorithm to optimize the problem.

The remainder of this article is organized as follows. Section II presents the overview of RDG. Section III gives the analysis on RDG and introduces our improved RDG. Section IV presents the experimental studies. Finally, Section V concludes this article.

II. Recursive Differential Grouping

Let X be the set of decision variables $\{x_1, \ldots, x_D\}$ and U_X be the set of unit vectors in the decision space $\mathbb{R}^D$. Let X_1 be a subset of X and U_{X_1} be a subset of U_X. For any unit vector $\mathbf{u} = (u_1, \ldots, u_D) \in U_{X_1}$, we have

$$u_i = 0, \text{ if } x_i \notin X_1. \tag{1}$$

The RDG [19] method examines the interrelationship between a pair of sets of variables not a pair of single variables according to the following theorem.

Theorem 1 [19]: Let $f : \mathbb{R}^D \to \mathbb{R}$ be an objective function; $X_1 \subset X$ and $X_2 \subset X$ be two mutually exclusive subsets of decision variables: $X_1 \cap X_2 = \emptyset$. If there exist two unit vectors $\mathbf{u}_1 \in U_{X_1}$ and $\mathbf{u}_2 \in U_{X_2}$, two real numbers $l_1, l_2 > 0$, and a candidate solution $\mathbf{x}^*$ in the decision space, such that

$$f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_2\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_2\big) \neq f\big(\mathbf{x}^* + l_1\mathbf{u}_1\big) - f\big(\mathbf{x}^*\big) \tag{2}$$

there is some interaction between decision variables in X_1 and X_2.

If (2) does not hold, RDG determines X_1 and X_2 are mutually separable sets. If X_1 and X_2 are interrelated with each other, RDG divides X_2 into two equal-sized and mutually exclusive subsets. RDG examines the interrelationship between X_1 and the two subsets. Repeat the above process until RDG finds the variables which interrelate with X_1. During this binary search, for the subset which X_1 does not interrelate with, RDG does not further examine the interrelationship between X_1 and this subset. This search branch is cut. It was reported in [19] that for decomposing a D-dimensional problem, the computational complexity of RDG is $\mathcal{O}(D \log_2 D)$. The experimental results in [19] showed that RDG can save more computation than DG [17], DG2 [18], and FII [22] as the dimensionality of a problem increases.

For brevity, the left-hand side of (2) is denoted by Δ_1 and the right-hand side is denoted by Δ_2. RDG uses $\lambda = |\Delta_1 - \Delta_2| > \epsilon$ to determine (2) holds. The performance of RDG on identifying the interrelationship between variables is sensitive to the value of ϵ [19], [21]. For different problems, the suitable values of ϵ may be different for identifying the interrelationship. The GDG method [23] for adapting the value of ϵ is adopted by RDG. This adaptation method may be unsuitable for decomposing imbalanced problems [18]. Inspired by DG2 [18], RDG2 [21] adapts the value of ϵ based on the computational round-off errors of RDG. An upper bound of the round-off errors is derived by RDG2 to set the value of ϵ. The experimental results in [21] showed that RDG2 improves the accuracy of RDG in identifying the interrelationship between variables.

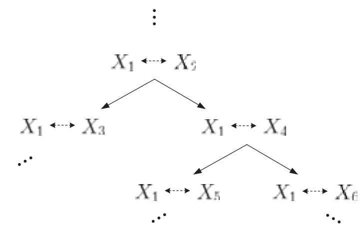

Fig. 1. Decomposition process of RDG, where X_i, $i = 1, \ldots, 6$ are sets of variables. $X_2 = X_3 \cup X_4$, $X_4 = X_5 \cup X_6$, and $X_3 \cap X_4 = X_5 \cap X_6 = \emptyset$.

III. Efficient Recursive Differential Grouping

In the section, we analyze the interrelationship examination process of RDG. We discover that the interrelationship examinations exist association during the process. Based on this discovery, an improved RDG named efficient RDG (ERDG) is proposed in this section. By exploiting the historical information on the interrelationship examination, ERDG can save redundant interrelationship examinations. The computational cost of ERDG is also analyzed in this section.

A. Analysis on Recursive Differential Grouping

Suppose X_1 and X_2 are two mutually exclusive subsets of variables. According to Theorem 1, to examine the interrelationship between X_1 and X_2, RDG computes Δ_1 and Δ_2 as follows:

$$\begin{aligned} \Delta_1 &= f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) \\ &\quad - f\big(\mathbf{x}^* + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) \\ \Delta_2 &= f\big(\mathbf{x}^* + l_1\mathbf{u}_1\big) - f\big(\mathbf{x}^*\big). \end{aligned} \tag{3}$$

If X_1 interrelates with X_2, RDG divides X_2 into two equal-sized and mutually exclusive subsets X_3 and X_4 (see Fig. 1). To examine the interrelationship between X_1 and X_3, RDG computes Δ_1' and Δ_2' as follows:

$$\begin{aligned} \Delta_1' &= f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_3\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_3\big) \\ \Delta_2' &= \Delta_2. \end{aligned} \tag{4}$$

To examine the interrelationship between X_1 and X_4, RDG computes Δ_1'' and Δ_2'' as follows:

$$\begin{aligned} \Delta_1'' &= f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_4\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_4\big) \\ \Delta_2'' &= \Delta_2. \end{aligned} \tag{5}$$

Because $X_2 = X_3 \cup X_4$, there may exist association between the above interrelationship examinations. Based on the previous interrelationship examinations (i.e., $\Delta_1 - \Delta_2$ and $\Delta_1' - \Delta_2'$), we can determine the interrelationship between X_1 and X_4 but not need to compute Δ_1''?

Proposition 1: If $(\Delta_1 - \Delta_2) = (\Delta_1' - \Delta_2')$, X_1 does not interrelate with X_4; otherwise, X_1 interrelates with X_4.

Proof: If $(\Delta_1 - \Delta_2) = (\Delta_1' - \Delta_2')$, since $\Delta_2 = \Delta_2'$, it is clear that $\Delta_1 = \Delta_1'$

$$\begin{aligned} &f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) - f\big(\mathbf{x}^* + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) \\ &\quad = f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_3\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_3\big). \end{aligned} \tag{6}$$

Let $\mathbf{x}' = \mathbf{x}^* + l_2\mathbf{u}_3$. Then, (6) is

$$\begin{aligned} &f\big(\mathbf{x}' + l_1\mathbf{u}_1 + l_2\mathbf{u}_4\big) - f\big(\mathbf{x}' + l_2\mathbf{u}_4\big) \\ &\quad = f\big(\mathbf{x}' + l_1\mathbf{u}_1\big) - f\big(\mathbf{x}'\big). \end{aligned} \tag{7}$$

Equation (7) indicates that X_1 does not interrelate with X_4.

If $(\Delta_1 - \Delta_2) \neq (\Delta_1' - \Delta_2')$, since $\Delta_2 = \Delta_2'$, it is clear that $\Delta_1 \neq \Delta_1'$

$$\begin{aligned} &f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) - f\big(\mathbf{x}^* + l_2(\mathbf{u}_3 + \mathbf{u}_4)\big) \\ &\quad \neq f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_3\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_3\big). \end{aligned} \tag{8}$$

Let $\mathbf{x}' = \mathbf{x}^* + l_2\mathbf{u}_3$. Then, (8) is

$$\begin{aligned} &f\big(\mathbf{x}' + l_1\mathbf{u}_1 + l_2\mathbf{u}_4\big) - f\big(\mathbf{x}' + l_2\mathbf{u}_4\big) \\ &\quad \neq f\big(\mathbf{x}' + l_1\mathbf{u}_1\big) - f\big(\mathbf{x}'\big). \end{aligned} \tag{9}$$

Equation (9) indicates that X_1 interrelates with X_4. ■

According to Proposition 1, based on the interrelationship examinations between X_1 and both of X_2 and X_3, we can determine whether X_1 interrelates with X_4 but avoid examining the interrelationship between X_1 and X_4, i.e., not need to compute Δ_1'' in (5).

B. Reducing Computational Cost

During the binary search of RDG, the interrelationship between X_1 and both of X_3 and X_4 can be categorized into the following three cases.

1) X_1 interrelates with X_3 but not interrelate with X_4: $(\Delta_1 - \Delta_2) = (\Delta_1' - \Delta_2')$.
2) X_1 does not interrelate with X_3 but interrelates with X_4: $(\Delta_1 - \Delta_2) \neq (\Delta_1' - \Delta_2')$.
3) X_1 interrelates with X_3 and X_4: $(\Delta_1 - \Delta_2) \neq (\Delta_1' - \Delta_2')$.

Note that if X_1 does not interrelate with X_2, RDG does not further examine the interrelationship between X_1 and X_2. Therefore, the case that X_1 does not interrelate with X_3 and X_4 does not occur during the binary search of RDG.

For the first case, according to Proposition 1, we can determine X_1 does not interrelate with X_4 but avoid the interrelationship examination, i.e., avoid computing Δ_1'' in (5). Because X_1 does not interrelate with X_4, the interrelationship between X_1 and X_4 will not be further examined. This search branch is cut. For the second and the third cases (X_1 interrelates with X_4), RDG continues with the binary search for interrelationship. In the above two cases, RDG divides X_4 into two equal-sized and mutually exclusive subsets X_5 and X_6. The interrelationship between X_1 and both of X_5 and X_6 is further examined (see Fig. 1). For this further examination, the value of $\Delta_1'' = f(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_4) - f(\mathbf{x}^* + l_2\mathbf{u}_4)$ should be known before Proposition 1 is applied.

In the interrelationship examination between X_1 and X_2, Δ_1 is computed [see (3)]. Let $\mathbf{x}' = \mathbf{x}^* + l_2\mathbf{u}_4$. Then

$$\Delta_1 = f\big(\mathbf{x}' + l_1\mathbf{u}_1 + l_2\mathbf{u}_3\big) - f\big(\mathbf{x}' + l_2\mathbf{u}_3\big). \tag{10}$$

Algorithm 1 (*sep*, *nonsep*) ← ERDG(f, **ub**, **lb**)

1: $sep, nonsep \leftarrow \emptyset$;
2: $\mathbf{x}_{l,l} \leftarrow \mathbf{lb}$; $y_{l,l} \leftarrow f(\mathbf{x}_{l,l})$;
3: $X_1 \leftarrow \{x_1\}$; $X_2 \leftarrow \{x_2, \ldots, x_D\}$;
4: **while** $X_2 \neq \emptyset$ **do**
5: $\quad \mathbf{x}_{u,l} \leftarrow \mathbf{x}_{l,l}$; $\mathbf{x}_{u,l}(X_1) \leftarrow \mathbf{ub}(X_1)$; $y_{u,l} \leftarrow f(\mathbf{x}_{u,l})$;
6: $\quad F \leftarrow \{y_{l,l}, y_{u,l}, nan, nan\}$;
7: $\quad (X_1^*, \hat{\beta}) \leftarrow$ INTERACT(X_1, X_2, $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, $\mathbf{ub}$, $\mathbf{lb}$, F);
8: $\quad$ **if** $|X_1^*| = |X_1|$ **then**
9: $\quad\quad$ **if** $|X_1^*| > 1$ **then**
10: $\quad\quad\quad nonsep \leftarrow \{nonsep, X_1^*\}$;
11: $\quad\quad$ **else**
12: $\quad\quad\quad sep \leftarrow sep \bigcup X_1^*$;
13: $\quad\quad$ **end if**
14: $\quad\quad X_1 \leftarrow \{x\}$ and $X_2 \leftarrow (X_2 - \{x\})$, where x is the first variable in X_2;
15: $\quad$ **else**
16: $\quad\quad X_1 \leftarrow X_1^*$; $X_2 \leftarrow (X_2 - X_1)$;
17: $\quad$ **end if**
18: $\quad$ **if** $X_2 = \emptyset$ **then**
19: $\quad\quad$ **if** $|X_1| > 1$ **then**
20: $\quad\quad\quad nonsep \leftarrow \{nonsep, X_1\}$;
21: $\quad\quad$ **else**
22: $\quad\quad\quad sep \leftarrow sep \bigcup X_1$;
23: $\quad\quad$ **end if**
24: $\quad$ **end if**
25: **end while**
26: return *sep* and *nonsep*;

If X_1 does not interrelate with X_3

$$\begin{aligned} \Delta_1 &= f\big(\mathbf{x}' + l_1\mathbf{u}_1 + l_2\mathbf{u}_3\big) - f\big(\mathbf{x}' + l_2\mathbf{u}_3\big) \\ &= f\big(\mathbf{x}' + l_1\mathbf{u}_1\big) - f\big(\mathbf{x}'\big) \\ &= f\big(\mathbf{x}^* + l_1\mathbf{u}_1 + l_2\mathbf{u}_4\big) - f\big(\mathbf{x}^* + l_2\mathbf{u}_4\big). \end{aligned} \tag{11}$$

When we determine the interrelationship between X_1 and X_6 according to Proposition 1, (11) shows that in the second case, we can use the equivalent Δ_1 to replace Δ_1'', which can save computational cost. In the third case, for this further examination, we still need to compute Δ_1'' before Proposition 1 is applied.

Among the above three cases, there are two cases (i.e., the first and the second cases) where we can determine the interrelationship between X_1 and X_4 based on the previous interrelationship examinations.

Algorithm 1 illustrates our proposed efficient RDG method named ERDG, where **ub** and **lb** are the upper and the lower bounds of the decision variables of decomposed problem f, respectively. ERDG first examines the interrelationship between the first variable x_1 and the remaining variables (i.e., X_1 and X_2). If there does not exist interrelationship between X_1 and X_2 (see step 8), X_1 and X_2 are mutually separable. In this case, if X_1 contains only one variable, ERDG puts the variable in *sep*; otherwise, the variables in X_1 are grouped as a subcomponent of nonseparable variables and ERDG puts X_1 in *nonsep*. If there exists interrelationship between X_1 and X_2, ERDG puts the variables which interrelate with X_1 into X_1 and deletes these variables from X_2 (see step 16). Repeat the above process until the interrelationship among all the variables is examined (i.e., X_2 is empty). ERDG adopts Algorithm 2 to examine the interrelationship between X_1 and X_2 (see step 7).

Algorithm 2 illustrates the method for the interrelationship examination between X_1 and X_2. If (2) holds, ERDG determines that there exists interrelationship between X_1 and X_2

Algorithm 2 $(X_1, \hat{\beta}) \leftarrow$ INTERACT(X_1, X_2, $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, **ub**, **lb**, F)

// Let F_1, F_2, F_3, and F_4 be the elements of F: $F = \{F_1, F_2, F_3, F_4\}$;
1: *nonSep* $\leftarrow 1$;
2: **if** $F_3 = nan$ **then** // *nan* is a non-numeric value.
3: $\quad \mathbf{x}_{m,l} \leftarrow \mathbf{x}_{l,l}$; $\mathbf{x}_{m,l}(X_2) \leftarrow (\mathbf{lb}(X_2) + \mathbf{ub}(X_2))/2$;
4: $\quad \mathbf{x}_{u,m} \leftarrow \mathbf{x}_{u,l}$; $\mathbf{x}_{u,m}(X_2) \leftarrow (\mathbf{lb}(X_2) + \mathbf{ub}(X_2))/2$;
5: $\quad F_3 \leftarrow f(\mathbf{x}_{m,l})$; $F_4 \leftarrow f(\mathbf{x}_{u,m})$;
6: $\quad \Delta_1 \leftarrow (F_1 - F_2)$; $\Delta_2 \leftarrow (F_3 - F_4)$; $\beta \leftarrow (\Delta_1 - \Delta_2)$;
7: $\quad$ **if** $|\beta| \le \epsilon$ **then**
8: $\quad\quad$ *nonSep* $\leftarrow 0$;
9: $\quad$ **end if**
10: **end if**
11: **if** *nonSep* $= 1$ **then**
12: $\quad$ **if** $|X_2| > 1$ **then**
13: $\quad\quad$ Divide X_2 into equal-sized and mutually exclusive subsets X_2' and X_2'';
14: $\quad\quad (X_1', \hat{\beta}) \leftarrow$ INTERACT(X_1, X_2', $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, **ub**, **lb**, $\{F_1, F_2, nan, nan\}$);
15: $\quad\quad$ **if** $\beta \ne \hat{\beta}$ **then**
16: $\quad\quad\quad$ **if** $|X_1'| = |X_1|$ **then**
17: $\quad\quad\quad\quad (X_1'', \beta') \leftarrow$ INTERACT(X_1, X_2'', $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, **ub**, **lb**, F);
18: $\quad\quad\quad$ **else**
19: $\quad\quad\quad\quad (X_1'', \beta') \leftarrow$ INTERACT(X_1, X_2'', $\mathbf{x}_{l,l}$, $\mathbf{x}_{u,l}$, **ub**, **lb**, $\{F_1, F_2, nan, nan\}$);
20: $\quad\quad\quad$ **end if**
21: $\quad\quad\quad X_1 \leftarrow X_1' \bigcup X_1''$;
22: $\quad\quad$ **else**
23: $\quad\quad\quad X_1 \leftarrow X_1'$;
24: $\quad\quad$ **end if**
25: $\quad$ **else**
26: $\quad\quad X_1 \leftarrow X_1 \bigcup X_2$;
27: $\quad$ **end if**
28: **end if**
29: return X_1 and β;

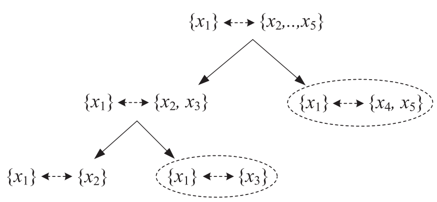

Fig. 2. Decomposition processes of RDG, RDG2, and ERDG on a partially separable function. The two-way dash arrow represents examining the interrelationship between the two sets of variables. ERDG does not examine the interrelationship with dash ellipse but RDG and RDG2 do.

and divides X_2 into two equal-sized subsets (see step 13). The interrelationship between X_1 and the two subsets is further examined. Repeat the above process until ERDG finds all the variables which interrelate with X_1.

After RDG divides X_2 into subsets X_2' and X_2'' (see step 13 in Algorithm 2), RDG spends the computation examining the interrelationship between X_1 and both of X_2' and X_2'', i.e., steps 14 and 19 in Algorithm 2. According to Proposition 1, if X_1 does not interrelate with X_2', ERDG can determine X_1 interrelates with X_2'' (see step 17 in Algorithm 2). In step 17, because $F_3 \ne nan$, the computation for examining the interrelationship between X_1 and X_2'', i.e., steps 3–9, is saved. Similarly, if X_1 interrelates with X_2' and $\beta = \hat{\beta}$, ERDG can determine that X_1 does not interrelate with X_2'' without examining the interrelationship (see step 23). Besides the above difference, ERDG moves an FE [i.e., $f(\mathbf{x}_{u,l})$] from Algorithm 2 to Algorithm 1, which does not affect the decomposition process. Therefore, ERDG spends two FEs each time the interrelationship is examined (see step 5 in Algorithm 2), while RDG or RDG2 spends three FEs [19], [21].

Take the following partially separable function as an example:

$$f(\mathbf{x}) = (x_1 - x_3)^2 + (x_2 - x_4)^2 + x_5^2.$$

$x_1 \leftrightarrow x_3$ and $x_2 \leftrightarrow x_4$, where $\leftrightarrow$ represents the two variables are interrelated with each other, and x_5 is a separable variable. Fig. 2 shows the binary searches of RDG, RDG2, and ERDG for the variable which interrelates with x_1. If (2) holds (i.e., there exists interrelationship between x_1 and the variables), RDG and RDG2 divide the variables into two equal-sized subsets of variables and continue with the search branch until finding the variable which interrelates with x_1 (i.e., x_3). If (2) does not hold, x_1 does not interrelate with the variables. RDG and RDG2 do not examine the branch further (e.g., the interrelationship examination between $\{x_1\}$ and $\{x_4, x_5\}$). According to Proposition 1, because $(\Delta_1 - \Delta_2) = (\Delta_1' - \Delta_2')$, ERDG determines that x_1 does not interrelate with x_4 and x_5 without examining the interrelationship between $\{x_1\}$ and $\{x_4, x_5\}$. Similarly, because x_1 interrelates with $\{x_2, x_3\}$ but not interrelates with x_2, ERDG determines x_1 interrelates with x_3 without examining the interrelationship between $\{x_1\}$ and $\{x_3\}$. The computational cost of ERDG is about half of the costs of RDG and RDG2.

C. Analysis on Computational Cost

Before the interrelationship examination starts, ERDG spends one FE (see step 2 in Algorithm 1). To find a subcomponent of variables, ERDG spends one FE (see step 5 in Algorithm 1) each time Algorithm 2 is invoked by Algorithm 1 (see step 7 in Algorithm 1). Two FEs are spent by ERDG each time the interrelationship is examined (see step 5 in Algorithm 2). Therefore, ERDG spends $2t+n+1$ FEs decomposing a problem, where t is the times the interrelationship is examined by ERDG and n is the times Algorithm 2 is invoked by Algorithm 1. Similarly, we can obtain that RDG and RDG2 spend $3t'+1$ FEs decomposing a problem, where t' is the times the interrelationship is examined by RDG and RDG2. Because ERDG can avoid examining some interrelationship, $t \le t'$.

For decomposing a D-dimensional problem, the theoretical computational complexity of ERDG is analyzed as follows.

1) For a fully separable problem, the interrelationship is examined $t = D - 1$ times and Algorithm 2 is invoked by Algorithm 1 $n = D - 1$ times. Therefore, ERDG spends $2(D-1)+D-1+1 = 3D-2$ FEs decomposing this kind of problem, which is the same with RDG and RDG2.
2) For a fully nonseparable problem, the decomposition process forms a binary tree. The number of all the nodes in a binary tree is $2N - 1$, where N is the number of leaf nodes. Therefore, for a D-dimensional fully nonseparable problem, the interrelationship is examined $t = 2(D - 1) - 1 = 2D - 3$ times. Algorithm 2 is invoked by Algorithm 1 $n = 1$ time. ERDG spends $2(2D-3)+1+1 = 4D-4$ FEs decomposing the fully nonseparable problem, while RDG or RDG2 spends $3t' + 1 = 3(2D - 3) + 1 = 6D - 8$ FEs.

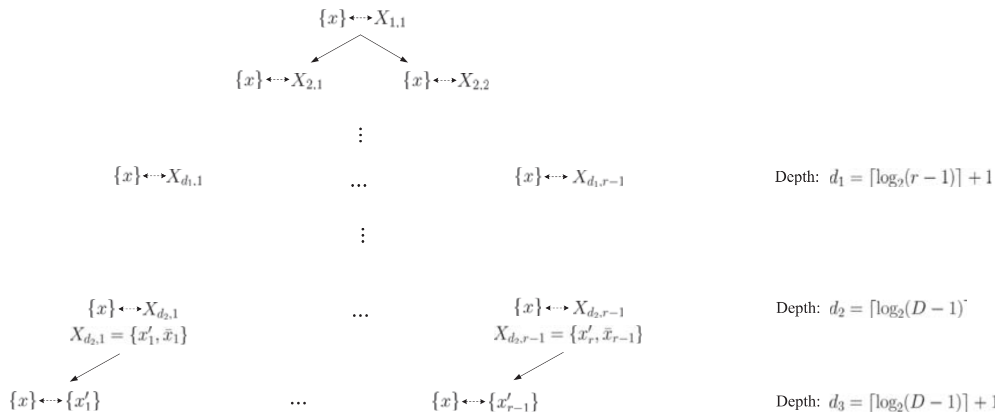

Fig. 3. Interrelationship examinations of ERDG on finding the variables which interrelate with x. The two-way dash arrow represents examining the interrelationship between two sets of variables.

3) For a partially separable problem with n' ($n' > 0$) subcomponents of nonseparable variables, where the sizes of the subcomponents are r_i ($r_i > 0$), $i = 1, \ldots, n'$, and s ($s \geq 0$) separable variables ($s + \sum_{i=1}^{n'} r_i = D$), the computational complexity of ERDG is analyzed as follows. Fig. 3 shows a decomposition process of ERDG for obtaining the subcomponent of the nonseparable variables including x, where r is the size of this subcomponent. When the depth of the binary tree is smaller than $d_1 = \lceil \log_2(r-1) \rceil + 1$, to find the $r - 1$ variables which interrelate with x, the nodes in the subtree are not more than $2^{\lceil \log_2(r-1) \rceil} - 1$, i.e., all the nodes at each level are examined. When the depth of the binary tree is not smaller than d_1, the nodes where X consists of only separable variables are not examined further and this search branch is cut. In this case, the nodes at each level are not more than $r - 1$. The depth of the binary tree is not larger than $\lceil \log_2(D-1) \rceil + 1$. Therefore, the number of the nodes in the binary tree in Fig. 3 (i.e., the times the interrelationship is examined) are not more than

$$2^{\lceil \log_2(r-1) \rceil} - 1 + \sum_{j=\lceil \log_2(r-1) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (r-1). \tag{12}$$

To obtain the n' subcomponents of nonseparable variables, the times the interrelationship is examined are not more than

$$t_1 = \sum_{i=1}^{n'} \left[2^{\lceil \log_2(r_i-1) \rceil} - 1 + \sum_{j=\lceil \log_2(r_i-1) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (r_i - 1) \right]. \tag{13}$$

$$\begin{aligned}
&2 \sum_{i=1}^{n'} \left[2^{\lceil \log_2(r_i-1) \rceil} - 1 + \sum_{j=\lceil \log_2(r_i-1) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (r_i - 1) \right] + 2s + n' + s + 1 \\
&= 2 \sum_{i=1}^{n'} \left(2^{\lceil \log_2(r_i-1) \rceil} \right) + 2 \sum_{i=1}^{n'} \sum_{j=\lceil \log_2(r_i-1) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (r_i - 1) + 3s - n' + 1 \\
&\approx 2 \sum_{i=1}^{n'} \log_2(r_i - 1) + 2 \sum_{i=1}^{n'} \left(\left(\log_2(D-1) + 1 - \log_2(r_i - 1) \right)(r_i - 1) \right) \\
&< 2D\log_2 D + 2D\log_2 D := \mathcal{O}(D \log_2 D)
\end{aligned} \tag{14}$$

$$\begin{aligned}
&3 \sum_{i=1}^{n'} \left[2^{\lceil \log_2(2(r_i-1)) \rceil} - 1 + \sum_{j=\lceil \log_2(2(r_i-1)) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (2(r_i - 1)) \right] + 3s + 1 \\
&= 6 \sum_{i=1}^{n'} \left(2^{\lceil \log_2(r_i-1) \rceil} \right) + 6 \sum_{i=1}^{n'} \sum_{j=\lceil \log_2(2(r_i-1)) \rceil + 1}^{\lceil \log_2(D-1) \rceil + 1} (r_i - 1) + 3(s - n') + 1 \\
&\approx 6 \sum_{i=1}^{n'} \log_2(r_i - 1) + 6 \sum_{i=1}^{n'} \left(\left(\log_2(D-1) + 1 - \log_2(2(r_i - 1)) \right)(r_i - 1) \right) \\
&< 6D\log_2 D + 6D\log_2 D := \mathcal{O}(D \log_2 D)
\end{aligned} \tag{15}$$

TABLE I
GROUPING RESULTS ON THE CEC'2013 FUNCTIONS. FOR EACH FUNCTION, THE BEST VALUES OF USED FEs AND GROUPING ACCURACIES AMONG DIFFERENT GROUPING METHODS ARE SHOWN IN BOLD FONT, RESPECTIVELY

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	1001000	**100.0%**	—	500501	**100.0%**	—	3008	**100.0%**	—	**2998**	**100.0%**	—	**2998**	**100.0%**	—
f_2	1001000	**100.0%**	—	500501	**100.0%**	—	3008	**100.0%**	—	**2998**	**100.0%**	—	**2998**	**100.0%**	—
f_3	1001000	**100.0%**	—	500501	0.0%	—	6005	0.0%	—	5992	0.0%	—	**3996**	0.0%	—
f_4	15706	5.4%	50.0%	500501	**100.0%**	**100.0%**	9842	**100.0%**	**100.0%**	9832	**100.0%**	**100.0%**	**5326**	**100.0%**	**100.0%**
f_5	527026	**100.0%**	66.7%	500501	**100.0%**	**100.0%**	10145	**100.0%**	**100.0%**	9895	**100.0%**	**100.0%**	**5395**	**100.0%**	**100.0%**
f_6	579848	**100.0%**	50.0%	500501	0.0%	**100.0%**	13574	3.6%	91.7%	11587	0.0%	**100.0%**	**5905**	0.0%	91.7%
f_7	11694	9.1%	0.0%	500501	**100.0%**	50.0%	11381	27.3%	0.0%	9814	**100.0%**	**100.0%**	**5554**	**100.0%**	**100.0%**
f_8	22682	—	65.0%	500501	—	**80.0%**	19364	—	70.0%	19405	—	**80.0%**	**8451**	—	75.0%
f_9	17650	—	**100.0%**	500501	—	**100.0%**	19343	—	**100.0%**	19156	—	**100.0%**	**8812**	—	**100.0%**
f_{10}	48650	—	65.0%	500501	—	**100.0%**	19178	—	85.0%	19879	—	**100.0%**	**8794**	—	87.5%
f_{11}	9332	—	0.0%	500501	—	**100.0%**	10496	—	0.0%	19429	—	**100.0%**	**9212**	—	**100.0%**
f_{12}	149894	—	0.0%	500501	—	**100.0%**	50876	—	**100.0%**	50866	—	**100.0%**	**26980**	—	**100.0%**
f_{13}	**5968**	—	0.0%	409966	—	**100.0%**	8345	—	0.0%	15187	—	0.0%	7599	—	0.0%
f_{14}	13968	—	0.0%	409966	—	**100.0%**	9542	—	**100.0%**	16150	—	**100.0%**	**8420**	—	**100.0%**
f_{15}	**2000**	—	**100.0%**	500501	—	**100.0%**	6173	—	**100.0%**	5992	—	**100.0%**	3996	—	**100.0%**
Total FEs	4,407,418			7,326,445			200,280			219,180			114,436		

Algorithm 2 is invoked by Algorithm 1 $n_1 = n'$ times. To obtain the s separable variables, the interrelationship is examined not more than $t_2 = s$ times and Algorithm 2 is invoked by Algorithm 1 not more than $n_2 = s$ times.[1] To decompose a partially separable problem with n' subcomponents of nonseparable variables and s separable variables, the maximum number of FEs used by ERDG is $2(t_1 + t_2) + n_1 + n_2 + 1$. Equation (14), as shown at the bottom of the previous page, shows its specific value. Similarly, we can obtain the maximum number of FEs used by RDG or RDG2 which is shown in (15), as shown at the bottom of the previous page. ERDG, RDG, and RDG2 have the same computational complexity, i.e., $\mathcal{O}(D \log_2 D)$. However, based on (14) and (15), as shown at the bottom of the previous page, we can estimate that ERDG can save up to about 2/3 of the computational cost of RDG and RDG2.

4) For a nonseparable problem with overlapping subcomponents, the n' ($n' > 0$) subcomponents of nonseparable variables are interrelated with each other via the overlapping variables. For a subcomponent of nonseparable variables, the decomposition process on this kind of problem is similar to the process on a partially separable problem. When ERDG completes obtaining the subcomponent of nonseparable variables, ERDG groups this subcomponent and the subcomponents overlapping with this subcomponent together. The computational complexity of ERDG for decomposing this kind of problem is similar to its computational complexity for decomposing a partially separable problem.

[1]At the end of the running of Algorithm 1, if X_1 consists of only one variable and $X_2 = \emptyset$, ERDG directly identifies this variable as a separable variable, i.e., Algorithm 1 does not invoke Algorithm 2. In this case, $t_2 = n_2 = s - 1$; otherwise, $t_2 = n_2 = s$.

TABLE II
EXTENDED CEC'2010 FUNCTIONS, WHERE THE NUMBER OF VARIABLES IN EACH SUBCOMPONENT OF NONSEPARABLE VARIABLES IN f_4–f_{18} IS FIXED TO 50, AS THE CONSISTENT SET IN THE ORIGINAL CEC'2010 BENCHMARK [24]

F	D	Sep	Nonsep	
		Number of Variables	Number of Subcomponents	Number of Variables
f_1–f_3	1000	1000	0	0
	2000	2000	0	0
	3000	3000	0	0
	4000	4000	0	0
	5000	5000	0	0
f_4–f_8	1000	950	1	50
	2000	1950	1	50
	3000	2950	1	50
	4000	3950	1	50
	5000	4950	1	50
f_9–f_{13}	1000	500	10	500
	2000	1000	20	1000
	3000	1500	30	1500
	4000	2000	40	2000
	5000	2500	50	2500
f_{14}–f_{18}	1000	0	20	1000
	2000	0	40	2000
	3000	0	60	3000
	4000	0	80	4000
	5000	0	100	5000
f_{19}–f_{20}	1000	0	1	1000
	2000	0	1	2000
	3000	0	1	3000
	4000	0	1	4000
	5000	0	1	5000

IV. EXPERIMENTAL STUDIES

A set of 35 test instances proposed in the IEEE CEC'2010 and CEC'2013 special sessions on large-scale global optimization was used to study the performance of ERDG. The detailed description of these test instances is given in [24] and [25]. ERDG was compared with DG [17],

TABLE III
GROUPING RESULTS ON THE 1000-D CEC'2010 FUNCTIONS. FOR EACH FUNCTION, THE BEST VALUES OF USED FEs AND GROUPING ACCURACIES AMONG DIFFERENT GROUPING METHODS ARE SHOWN IN BOLD FONT, RESPECTIVELY

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	1001000	**100.0%**	—	500501	**100.0%**	—	3008	**100.0%**	—	**2998**	**100.0%**	—	**2998**	**100.0%**	—
f_2	1001000	**100.0%**	—	500501	**100.0%**	—	3008	**100.0%**	—	**2998**	**100.0%**	—	**2998**	**100.0%**	—
f_3	1001000	**100.0%**	—	500501	0.0%	—	6002	0.0%	—	5992	0.0%	—	**3996**	0.0%	—
f_4	14554	3.5%	**100.0%**	500501	**100.0%**	**100.0%**	4208	**100.0%**	**100.0%**	4198	**100.0%**	**100.0%**	**3398**	**100.0%**	**100.0%**
f_5	905450	**100.0%**	**100.0%**	500501	**100.0%**	**100.0%**	4154	**100.0%**	**100.0%**	4144	**100.0%**	**100.0%**	**3458**	**100.0%**	**100.0%**
f_6	906332	**100.0%**	**100.0%**	500501	8.6%	**100.0%**	49880	**100.0%**	**100.0%**	8563	0.0%	**100.0%**	**5123**	0.0%	**100.0%**
f_7	67742	26.1%	0.0%	500501	**100.0%**	**100.0%**	4232	**100.0%**	**100.0%**	4222	**100.0%**	**100.0%**	**3432**	**100.0%**	**100.0%**
f_8	23286	14.0%	0.0%	500501	**100.0%**	**100.0%**	5609	**100.0%**	**100.0%**	5599	**100.0%**	**100.0%**	**4081**	**100.0%**	**100.0%**
f_9	270802	**100.0%**	**100.0%**	500501	**100.0%**	**100.0%**	14036	**100.0%**	**100.0%**	14026	**100.0%**	**100.0%**	**7226**	**100.0%**	**100.0%**
f_{10}	272958	**100.0%**	**100.0%**	500501	**100.0%**	**100.0%**	14018	**100.0%**	**100.0%**	14008	**100.0%**	**100.0%**	**7134**	**100.0%**	**100.0%**
f_{11}	270640	**100.0%**	90.0%	500501	0.0%	**100.0%**	13694	0.0%	**100.0%**	13684	0.0%	**100.0%**	**6798**	0.0%	**100.0%**
f_{12}	271390	**100.0%**	**100.0%**	500501	**100.0%**	**100.0%**	14318	**100.0%**	**100.0%**	14308	**100.0%**	**100.0%**	**7590**	**100.0%**	**100.0%**
f_{13}	50328	21.4%	0.0%	500501	**100.0%**	**100.0%**	29243	**100.0%**	**100.0%**	29233	**100.0%**	**100.0%**	**13165**	**100.0%**	**100.0%**
f_{14}	21000	—	**100.0%**	500501	—	**100.0%**	20564	—	**100.0%**	20554	—	**100.0%**	**9408**	—	**100.0%**
f_{15}	21000	—	**100.0%**	500501	—	**100.0%**	20522	—	**100.0%**	20512	—	**100.0%**	**9342**	—	**100.0%**
f_{16}	21128	—	80.0%	500501	—	**100.0%**	20918	—	**100.0%**	20908	—	**100.0%**	**9456**	—	**100.0%**
f_{17}	21000	—	**100.0%**	500501	—	**100.0%**	20768	—	**100.0%**	20758	—	**100.0%**	**9460**	—	**100.0%**
f_{18}	39624	—	0.0%	500501	—	**100.0%**	49862	—	**100.0%**	49852	—	**100.0%**	**21224**	—	**100.0%**
f_{19}	**2000**	—	**100.0%**	500501	—	**100.0%**	6002	—	**100.0%**	5992	—	**100.0%**	3996	—	**100.0%**
f_{20}	155430	—	0.0%	500501	—	**100.0%**	50876	—	**100.0%**	50866	—	**100.0%**	**27648**	—	**100.0%**
Total FEs	6,337,664			10,010,020			354,922			313,415			161,931		

DG2 [18], RDG [19], and RDG2 [21]. The parameters of the grouping methods were set as in their publications. ERDG set ϵ as RDG2 did.

Two metrics were used to evaluate the performance of the grouping methods: the number of FEs used to decompose a problem and the grouping accuracy. The smaller the number of FEs is and the higher the grouping accuracy is, the better the performance of a grouping method is. For a problem, let sep' denote the set of true separable variables and $\mathrm{nonsep}' = \{g'_1, \ldots, g'_{n'}\}$ denote the groups of true nonseparable variables. For $g'_i, i = 1, \ldots, n'$, each g'_i is a set of nonseparable variables and all the g'_i are mutually separable. sep and $\mathrm{nonsep} = \{g_1, \ldots, g_{\hat{n}}\}$, which have similar meanings to sep' and nonsep', respectively, denote the grouping result of a grouping method. The grouping accuracy of a grouping method is defined as follows.

1) For separable variables, the grouping accuracy is

$$\frac{|\mathrm{sep} \cap \mathrm{sep}'|}{|sep'|}.$$

2) For nonseparable variables, the grouping accuracy is

$$\frac{\sum_{g'_i \in \mathrm{nonsep}} |g'_i|}{\sum_{g'_i \in \mathrm{nonsep}'} |g'_i|}.$$

A. Comparison on Decomposition

1) Comparison on the IEEE CEC'2013 Functions: The CEC'2013 functions are classified into the following five categories.

1) Fully separable functions (f_1–f_3).
2) Partially separable functions with seven subcomponents of nonseparable variables and 700 separable variables (f_4–f_7).
3) Partially separable functions with 20 subcomponents of nonseparable variables (f_8–f_{11}).
4) Nonseparable functions with overlapping subcomponents (f_{12}–f_{14}).
5) Fully nonseparable function (f_{15}).

Table I summarizes the grouping results on the CEC'2013 functions. The percentage of the FEs saved by ERDG can be seen in Section I in the supplementary material accompanying this article. ERDG correctly decomposes all the functions except f_3, f_6, f_8, f_{10}, and f_{13}. ERDG performs better than RDG on all the functions in terms of the grouping accuracy. Theoretically, the grouping accuracies of ERDG should be the same with the ones of RDG2, but ERDG performs slightly worse than RDG2 on three functions (i.e., f_6, f_8, and f_{10}), which results from the computational round-off errors caused in the practical execution of the improvement in Section III-B. Overall, DG and DG2 use much more FEs than RDG, RDG2, and ERDG. It can be seen in Table I that ERDG saves about half of the FEs used by RDG or RDG2 on most of the functions. ERDG uses the fewest FEs on 14 out of 15 functions. This is because ERDG can avoid examining some interrelationship during the binary search for interrelationship and ERDG moves am FE out from the recursive process (see Section III-B). The FEs reduced by the two improvements can be seen in Section II in the supplementary material accompanying this article.

2) Comparison on High-Dimensional Functions: The structure of the CEC'2013 functions is fixed. Therefore, we used the CEC'2010 functions with the dimensionality of $D = \{1000,$

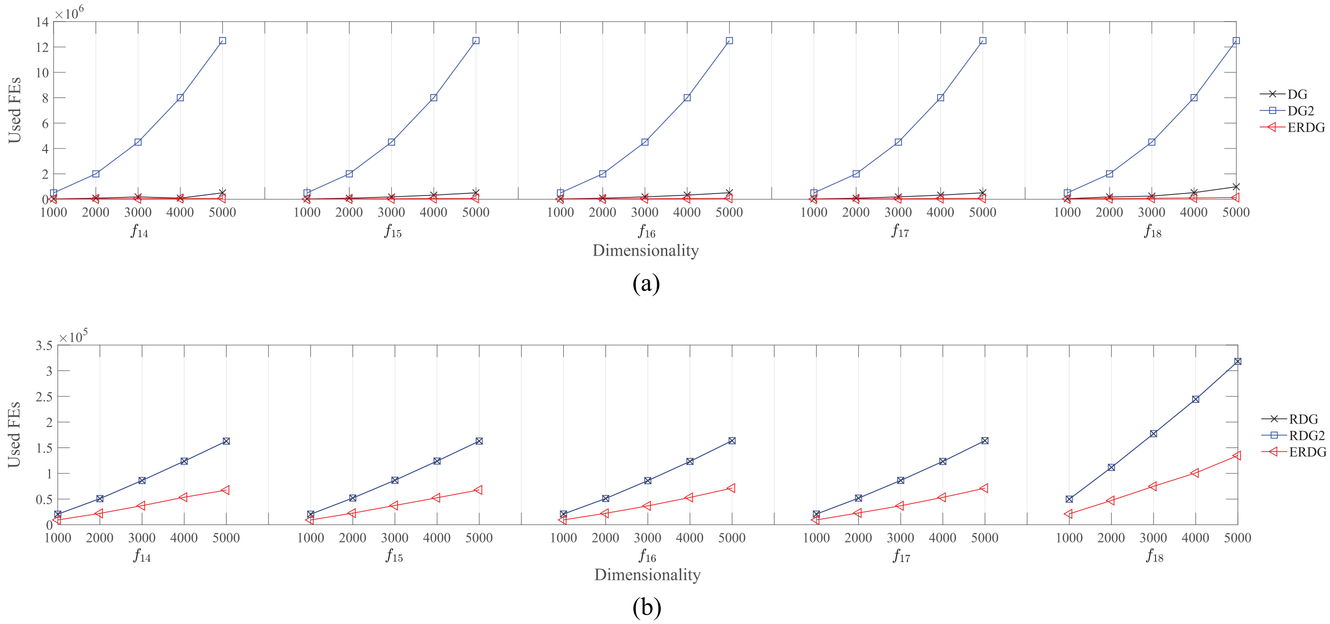

Fig. 4. Used FEs of the grouping methods on the selected CEC'2010 functions f_{14}–f_{18}. (a) ERDG versus DG and DG2. (b) ERDG versus RDG and RDG2.

2000, 3000, 4000, 5000} to investigate the performance of ERDG on the high-dimensional functions. The CEC'2010 functions are classified into the following categories.

1) Fully separable functions (f_1–f_3).
2) Partially separable functions (f_4–f_{18}).
3) Nonseparable function with overlapping subcomponents (f_{19}).
4) Fully nonseparable function (f_{20}).

Table II summarizes the extended CEC'2010 functions. As the dimensionality of these functions increases, there are more separable variables and subcomponents of nonseparable variables.

Tables III–V summarize the grouping results on the CEC'2010 functions with the dimensionality of $D = \{1000, 2000, 3000, 4000, 5000\}$. The percentage of the FEs saved by ERDG can be seen in Section I in the supplementary material accompanying this article. ERDG correctly decomposes almost all the functions. Although DG2 performs slightly better than ERDG on f_6 in terms of grouping accuracy, DG2 uses much more FEs than ERDG to decompose a function. ERDG uses the fewest FEs on 19 out of 20 1000-D CEC'2010 functions. ERDG saves about half of the FEs used by RDG or RDG2 on most of the functions. It can be seen in Tables III–V that the grouping methods spend more FEs decomposing the functions as the dimensionality of the functions increases. Especially, the numbers of FEs used by DG and DG2 increase more rapidly than the numbers of FEs used by RDG, RDG2, and ERDG. ERDG saves more FEs than RDG and RDG2 on the CEC'2010 functions as the dimensionality of the functions increases. Overall, ERDG spends the fewest FEs decomposing almost all the CEC'2010 function. The FEs reduced by the improvements of ERDG can be seen in Section II in the supplementary material accompanying this article.

Fig. 4 shows the FEs used by the grouping methods on the partially separable CEC'2010 functions f_{14}–f_{18} with the dimensionality of $D = \{1000, 2000, 3000, 4000, 5000\}$. The computational complexity of DG2 is $\mathcal{O}(D^2)$, while the computational complexity of the RDG methods (i.e., RDG, RDG2, and ERDG) is $\mathcal{O}(D \log_2 D)$. It can be seen in Fig. 4 that DG2 uses much more FEs than the other grouping methods and the number of FEs used by DG2 increases much more rapidly as the dimensionality increases, while the numbers of FEs used by RDG, RDG2, and ERDG approximately increase linearly with the dimensionality. Compared with RDG and RDG2, ERDG saves more FEs as the dimensionality increases.

B. Comparison on Optimization

In this section, we used the CEC'2013 functions to test whether ERDG can improve the performance of CCFR [26] on solving large-scale optimization problems. CCFR, a recently proposed contribution-based CC, is an efficient CC for solving large-scale optimization problems. CCFR adopted CMA-ES [27] as the optimizer. The parameters of CCFR were set as in its publication. We set the maximum number of fitness evaluations to 3×10^6 as the termination criterion for running an algorithm, which is suggested in [25]. The fitness evaluations spent by the grouping methods are counted as part of the computational budget.

Table VI summarizes the results of CCFR with different grouping methods on optimizing the CEC'2013 functions. For the fully separable functions f_1–f_3 and the partially separable functions f_4–f_{11}, because CCFR adopts the divide-and-conquer strategy to solve these functions, CCFR with the grouping methods except DG performs significantly better than CMA-ES which optimizes all the decision variables together. CCFR with DG performs worse than CMA-ES on these functions, which results from that DG cannot correctly group the nonseparable variables together (see Table I). A high grouping accuracy is crucial to the efficiency of the divide-and-conquer strategy of CC. Because the grouping accuracies of ERDG are

TABLE IV

GROUPING RESULTS ON THE 2000-D AND 3000-D CEC'2010 FUNCTIONS. FOR EACH FUNCTION, THE BEST VALUES OF USED FEs AND GROUPING ACCURACIES AMONG DIFFERENT GROUPING METHODS ARE SHOWN IN BOLD FONT, RESPECTIVELY

$D = 2000$

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	4002000	**100.0%**	—	2001001	**100.0%**	—	6008	**100.0%**	—	**5998**	**100.0%**	—	**5998**	**100.0%**	—
f_2	4002000	**100.0%**	—	2001001	**100.0%**	—	6008	**100.0%**	—	**5998**	**100.0%**	—	**5998**	**100.0%**	—
f_3	4002000	**100.0%**	—	2001001	0.0%	—	12005	0.0%	—	11992	0.0%	—	**7996**	0.0%	—
f_4	33436	3.5%	0.0%	2001001	**100.0%**	**100.0%**	7442	**100.0%**	**100.0%**	7432	**100.0%**	**100.0%**	**6476**	**100.0%**	**100.0%**
f_5	3810214	**100.0%**	**100.0%**	2001001	**100.0%**	**100.0%**	7538	**100.0%**	**100.0%**	7528	**100.0%**	**100.0%**	**6554**	**100.0%**	**100.0%**
f_6	3808450	**100.0%**	**100.0%**	2001001	22.1%	**100.0%**	53198	**100.0%**	**100.0%**	2205754	**100.0%**	**100.0%**	**10333**	0.4%	**100.0%**
f_7	31998	5.9%	**100.0%**	2001001	**100.0%**	**100.0%**	7460	**100.0%**	**100.0%**	7450	**100.0%**	**100.0%**	**6482**	**100.0%**	**100.0%**
f_8	27978	4.2%	0.0%	2001001	**100.0%**	**100.0%**	9068	**100.0%**	**100.0%**	9058	**100.0%**	**100.0%**	**7280**	**100.0%**	**100.0%**
f_9	1096986	**100.0%**	**100.0%**	2001001	**100.0%**	**100.0%**	33608	**100.0%**	**100.0%**	33598	**100.0%**	**100.0%**	**16770**	**100.0%**	**100.0%**
f_{10}	1103650	**100.0%**	**100.0%**	2001001	**100.0%**	**100.0%**	33416	**100.0%**	**100.0%**	33406	**100.0%**	**100.0%**	**16790**	**100.0%**	**100.0%**
f_{11}	1096872	**100.0%**	85.0%	2001001	0.0%	**100.0%**	30836	0.0%	**100.0%**	30826	0.0%	**100.0%**	**14774**	0.0%	**100.0%**
f_{12}	1088754	**100.0%**	**100.0%**	2001001	**100.0%**	**100.0%**	33410	**100.0%**	**100.0%**	33400	**100.0%**	**100.0%**	**16310**	**100.0%**	**100.0%**
f_{13}	137798	18.4%	0.0%	2001001	**100.0%**	**100.0%**	64448	**100.0%**	**100.0%**	64438	**100.0%**	**100.0%**	**28632**	**100.0%**	**100.0%**
f_{14}	82000	—	**100.0%**	2001001	—	**100.0%**	50846	—	**100.0%**	50836	—	**100.0%**	**22162**	—	**100.0%**
f_{15}	82000	—	**100.0%**	2001001	—	**100.0%**	52064	—	**100.0%**	52054	—	**100.0%**	**22636**	—	**100.0%**
f_{16}	82256	—	87.5%	2001001	—	**100.0%**	51128	—	**100.0%**	51118	—	**100.0%**	**22302**	—	**100.0%**
f_{17}	82000	—	**100.0%**	2001001	—	**100.0%**	51854	—	**100.0%**	51844	—	**100.0%**	**22966**	—	**100.0%**
f_{18}	181776	—	0.0%	2001001	—	**100.0%**	111731	—	**100.0%**	111721	—	**100.0%**	**47289**	—	**100.0%**
f_{19}	**4000**	—	**100.0%**	2001001	—	**100.0%**	12002	—	**100.0%**	11992	—	**100.0%**	7996	—	**100.0%**
f_{20}	582720	—	0.0%	2001001	—	**100.0%**	113732	—	**100.0%**	113722	—	**100.0%**	**60964**	—	**100.0%**
Total FEs	25,338,888			40,020,020			747,802			2,900,165			356,708		

$D = 3000$

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	9003000	**100.0%**	—	4501501	**100.0%**	—	9008	**100.0%**	—	**8998**	**100.0%**	—	**8998**	**100.0%**	—
f_2	9003000	**100.0%**	—	4501501	**100.0%**	—	9008	**100.0%**	—	**8998**	**100.0%**	—	**8998**	**100.0%**	—
f_3	9003000	**100.0%**	—	4501501	0.0%	—	18005	0.0%	—	17992	0.0%	—	**11996**	0.0%	—
f_4	37564	2.9%	0.0%	4501501	**100.0%**	**100.0%**	10634	**100.0%**	**100.0%**	10624	**100.0%**	**100.0%**	**9544**	**100.0%**	**100.0%**
f_5	8712920	**100.0%**	**100.0%**	4501501	**100.0%**	**100.0%**	10634	**100.0%**	**100.0%**	10624	**100.0%**	**100.0%**	**9588**	**100.0%**	**100.0%**
f_6	8712234	**100.0%**	**100.0%**	4501501	8.3%	**100.0%**	60302	**100.0%**	**100.0%**	1837384	**100.0%**	**100.0%**	**16704**	0.2%	0.0%
f_7	273976	16.7%	**100.0%**	4501501	**100.0%**	**100.0%**	10532	**100.0%**	**100.0%**	10522	**100.0%**	**100.0%**	**9508**	**100.0%**	**100.0%**
f_8	38738	3.6%	0.0%	4501501	**100.0%**	**100.0%**	12035	**100.0%**	**100.0%**	12025	**100.0%**	**100.0%**	**10141**	**100.0%**	**100.0%**
f_9	2450916	**100.0%**	**100.0%**	4501501	**100.0%**	**100.0%**	54854	**100.0%**	**100.0%**	54844	**100.0%**	**100.0%**	**26510**	**100.0%**	**100.0%**
f_{10}	2477768	**100.0%**	**100.0%**	4501501	**100.0%**	**100.0%**	55112	**100.0%**	**100.0%**	55102	**100.0%**	**100.0%**	**27126**	**100.0%**	**100.0%**
f_{11}	2530714	**100.0%**	70.0%	4501501	0.0%	**100.0%**	50921	0.0%	**100.0%**	50752	0.0%	**100.0%**	**24356**	0.0%	**100.0%**
f_{12}	2490312	**100.0%**	**100.0%**	4501501	**100.0%**	**100.0%**	55178	**100.0%**	**100.0%**	55168	**100.0%**	**100.0%**	**25430**	**100.0%**	**100.0%**
f_{13}	94892	4.9%	0.0%	4501501	**100.0%**	**100.0%**	101411	**100.0%**	**100.0%**	101401	**100.0%**	**100.0%**	**45393**	**100.0%**	**100.0%**
f_{14}	183000	—	**100.0%**	4501501	—	**100.0%**	86198	—	**100.0%**	86188	—	**100.0%**	**37158**	—	**100.0%**
f_{15}	183000	—	**100.0%**	4501501	—	**100.0%**	86642	—	**100.0%**	86632	—	**100.0%**	**37552**	—	**100.0%**
f_{16}	184358	—	78.3%	4501501	—	**100.0%**	85880	—	**100.0%**	85870	—	**100.0%**	**36752**	—	**100.0%**
f_{17}	183000	—	**100.0%**	4501501	—	**100.0%**	86348	—	**100.0%**	86338	—	**100.0%**	**37052**	—	**100.0%**
f_{18}	247254	—	0.0%	4501501	—	**100.0%**	177434	—	**100.0%**	177424	—	**100.0%**	**74876**	—	**100.0%**
f_{19}	**6000**	—	**100.0%**	4501501	—	**100.0%**	18002	—	**100.0%**	17992	—	**100.0%**	11996	—	**100.0%**
f_{20}	539254	—	0.0%	4501501	—	**100.0%**	182444	—	**100.0%**	182434	—	**100.0%**	**103502**	—	**100.0%**
Total FEs	56,354,900			90,030,020			1,180,582			2,957,312			573,180		

significantly higher than the ones of RDG on f_7, f_8, and f_{11}, CCFR with ERDG performs significantly better than CCFR with RDG by several orders of magnitude on the three functions. CCFR with ERDG performs significantly worse than CCFR with RDG2 on f_8 and f_{10} where the grouping accuracies of ERDG is worse than the ones of RDG2.

Although DG2 correctly groups the nonseparable variables together on most of the functions (see Table I), the results in Table VI show that the performance of CCFR with DG2 is significantly worse than CCFR with ERDG on seven functions, especially on f_7 and f_{11} where CCFR with ERDG performs better than CCFR with DG2 by several orders of magnitude.

TABLE V
GROUPING RESULTS ON THE 4000-D AND 5000-D CEC'2010 FUNCTIONS. FOR EACH FUNCTION, THE BEST VALUES OF USED FEs AND GROUPING ACCURACIES AMONG DIFFERENT GROUPING METHODS ARE SHOWN IN BOLD FONT, RESPECTIVELY

$D = 4000$

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	16004000	**100.0%**	—	8002001	**100.0%**	—	12008	**100.0%**	—	**11998**	**100.0%**	—	**11998**	**100.0%**	—
f_2	16004000	**100.0%**	—	8002001	**100.0%**	—	12008	**100.0%**	—	**11998**	**100.0%**	—	**11998**	**100.0%**	—
f_3	16004000	**100.0%**	—	8002001	0.0%	—	24029	0.0%	—	23992	0.0%	—	**15996**	0.0%	—
f_4	28712	2.1%	0.0%	8002001	**100.0%**	**100.0%**	13850	**100.0%**	**100.0%**	13840	**100.0%**	**100.0%**	**12690**	**100.0%**	**100.0%**
f_5	15628464	**100.0%**	**100.0%**	8002001	**100.0%**	**100.0%**	13784	**100.0%**	**100.0%**	13774	**100.0%**	**100.0%**	**12590**	**100.0%**	**100.0%**
f_6	15615136	**100.0%**	**100.0%**	8002001	15.7%	**100.0%**	64070	**100.0%**	**100.0%**	1547260	**100.0%**	**100.0%**	**22364**	0.3%	0.0%
f_7	554384	18.1%	**100.0%**	8002001	**100.0%**	**100.0%**	13802	**100.0%**	**100.0%**	13792	**100.0%**	**100.0%**	**12596**	**100.0%**	**100.0%**
f_8	1486788	30.5%	0.0%	8002001	**100.0%**	**100.0%**	15428	**100.0%**	**100.0%**	15418	**100.0%**	**100.0%**	**13324**	**100.0%**	**100.0%**
f_9	4361018	**100.0%**	77.5%	8002001	**100.0%**	**100.0%**	78146	**100.0%**	**100.0%**	78136	**100.0%**	**100.0%**	**36958**	**100.0%**	**100.0%**
f_{10}	4387570	**100.0%**	**100.0%**	8002001	99.8%	**100.0%**	77948	**100.0%**	**100.0%**	77938	**100.0%**	**100.0%**	**37834**	**100.0%**	**100.0%**
f_{11}	4385744	**100.0%**	82.5%	8002001	0.0%	**100.0%**	72863	0.0%	**100.0%**	71908	0.0%	**100.0%**	**33388**	0.0%	**100.0%**
f_{12}	4407268	**100.0%**	**100.0%**	8002001	**100.0%**	**100.0%**	77984	**100.0%**	**100.0%**	77974	**100.0%**	**100.0%**	**36532**	**100.0%**	**100.0%**
f_{13}	329330	9.0%	0.0%	8002001	**100.0%**	**100.0%**	140552	**100.0%**	**100.0%**	140542	**100.0%**	**100.0%**	**60868**	**100.0%**	**100.0%**
f_{14}	87228	—	15.0%	8002001	—	**100.0%**	123812	—	**100.0%**	123802	—	**100.0%**	**53604**	—	**100.0%**
f_{15}	324000	—	**100.0%**	8002001	—	**100.0%**	124034	—	**100.0%**	124024	—	**100.0%**	**52540**	—	**100.0%**
f_{16}	325580	—	81.3%	8002001	—	**100.0%**	123350	—	**100.0%**	123340	—	**100.0%**	**53154**	—	**100.0%**
f_{17}	324000	—	**100.0%**	8002001	—	**100.0%**	123356	—	**100.0%**	123346	—	**100.0%**	**53312**	—	**100.0%**
f_{18}	520362	—	1.3%	8002001	—	**100.0%**	244412	—	**100.0%**	244402	—	**100.0%**	**100930**	—	**100.0%**
f_{19}	**8000**	—	**100.0%**	8002001	—	**100.0%**	24002	—	**100.0%**	23992	—	**100.0%**	15996	—	**100.0%**
f_{20}	1720578	—	0.0%	8002001	—	**100.0%**	251444	—	**100.0%**	251434	—	**100.0%**	**126616**	—	**100.0%**
Total FEs	102,506,162			160,040,020			1,630,882			3,112,910			775,288		

$D = 5000$

F	DG			DG2			RDG			RDG2			ERDG		
	FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy		FEs	Accuracy	
		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep		Sep	Nonsep
f_1	24814006	96.8%	—	12502501	**100.0%**	—	15008	**100.0%**	—	**14998**	**100.0%**	—	**14998**	**100.0%**	—
f_2	25005000	**100.0%**	—	12502501	97.9%	—	15008	**100.0%**	—	**14998**	**100.0%**	—	**14998**	**100.0%**	—
f_3	25005000	**100.0%**	—	12502501	0.0%	—	30101	0.0%	—	29992	0.0%	—	**19996**	0.0%	—
f_4	60498	1.9%	0.0%	12502501	**100.0%**	**100.0%**	16838	**100.0%**	**100.0%**	16828	**100.0%**	**100.0%**	**15692**	**100.0%**	**100.0%**
f_5	24524506	**100.0%**	**100.0%**	12502501	**100.0%**	**100.0%**	16970	**100.0%**	**100.0%**	16960	**100.0%**	**100.0%**	**15652**	**100.0%**	**100.0%**
f_6	24524506	**100.0%**	**100.0%**	12502501	25.9%	**100.0%**	70538	**100.0%**	**100.0%**	1439020	**100.0%**	**100.0%**	**26937**	5.1%	0.0%
f_7	2435272	31.1%	**100.0%**	12502501	**100.0%**	**100.0%**	16838	**100.0%**	**100.0%**	16828	**100.0%**	**100.0%**	**15608**	**100.0%**	**100.0%**
f_8	89546	3.8%	0.0%	12502501	**100.0%**	**100.0%**	18332	**100.0%**	**100.0%**	18322	**100.0%**	**100.0%**	**16290**	**100.0%**	**100.0%**
f_9	6967806	**100.0%**	**100.0%**	12502501	**100.0%**	**100.0%**	101978	**100.0%**	**100.0%**	101968	**100.0%**	**100.0%**	**48758**	**100.0%**	**100.0%**
f_{10}	6848442	**100.0%**	**100.0%**	12502501	**100.0%**	**100.0%**	102176	**100.0%**	**100.0%**	102166	**100.0%**	**100.0%**	**48842**	**100.0%**	**100.0%**
f_{11}	6953454	**100.0%**	74.0%	12502501	0.0%	**100.0%**	94247	0.1%	**100.0%**	93694	0.0%	**100.0%**	**43474**	0.0%	**100.0%**
f_{12}	6887544	**100.0%**	**100.0%**	12502501	**100.0%**	**100.0%**	102146	**100.0%**	**100.0%**	102136	**100.0%**	**100.0%**	**46670**	**100.0%**	**100.0%**
f_{13}	595440	12.2%	0.0%	12502501	**100.0%**	**100.0%**	181604	**100.0%**	**100.0%**	181594	**100.0%**	**100.0%**	**79112**	**100.0%**	**100.0%**
f_{14}	505000	—	**100.0%**	12502501	—	**100.0%**	162842	—	**100.0%**	162832	—	**100.0%**	**67484**	—	**100.0%**
f_{15}	505000	—	**100.0%**	12502501	—	**100.0%**	162950	—	**100.0%**	162940	—	**100.0%**	**67882**	—	**100.0%**
f_{16}	511420	—	80.0%	12502501	—	**100.0%**	163706	—	**100.0%**	163696	—	**100.0%**	**71414**	—	**100.0%**
f_{17}	505000	—	**100.0%**	12502501	—	**100.0%**	163838	—	**100.0%**	163828	—	**100.0%**	**71218**	—	**100.0%**
f_{18}	981846	—	0.0%	12502501	—	**100.0%**	318104	—	**100.0%**	318094	—	**100.0%**	**134836**	—	**100.0%**
f_{19}	**10000**	—	**100.0%**	12502501	—	**100.0%**	30002	—	**100.0%**	29992	—	**100.0%**	19996	—	**100.0%**
f_{20}	1135178	—	0.0%	12502501	—	**100.0%**	325868	—	**100.0%**	325858	—	**100.0%**	**192106**	—	**100.0%**
Total FEs	158,864,464			250,050,020			2,109,094			3,476,744			1,031,963		

This is because DG2 spends much more FEs grouping the variables than ERDG (see Table I), which results in that CCFR with DG2 uses fewer FEs for optimization than CCFR with ERDG. ERDG uses fewer FEs than RDG and RDG2 to decompose a function, CCFR with ERDG saves more FEs for optimizing the function than CCFR with RDG and RDG2. Therefore, CCFR with ERDG outperforms CCFR with RDG and RDG2 on optimizing the CEC'2013 functions.

TABLE VI

AVERAGE FITNESS VALUES ± STANDARD DEVIATIONS ON THE CEC'2013 FUNCTIONS OVER 25 INDEPENDENT RUNS. THE SIGNIFICANTLY BEST RESULTS ARE IN BOLD FONT (WILCOXON TEST WITH HOLM p-VALUE CORRECTION, $\alpha = 0.05$). b, n, AND l DENOTE THE NUMBER OF FUNCTIONS WHERE CCFR–ERDG–CMA-ES PERFORMS SIGNIFICANTLY BETTER THAN, STATISTICALLY EQUIVALENT TO, AND SIGNIFICANTLY WORSE THAN ITS COMPETITORS, RESPECTIVELY

F	CCFR-ERDG-CMA-ES	CCFR-RDG2-CMA-ES	CCFR-RDG-CMA-ES	CCFR-DG2-CMA-ES	CCFR-DG-CMA-ES	CMA-ES
f_1	6.07e-17±5.34e-18	6.07e-17±5.34e-18	6.07e-17±5.34e-18	6.07e-17±5.34e-18	6.07e-17±5.34e-18	2.78e+05±2.55e+04↑
f_2	4.58e+02±3.68e+01	4.58e+02±3.68e+01	4.58e+02±3.68e+01	4.58e+02±3.68e+01	4.58e+02±3.68e+01	4.81e+03±1.95e+02↑
f_3	2.04e+01±5.45e-02	2.04e+01±5.45e-02↑	2.04e+01±5.45e-02↑	2.04e+01±4.90e-02↑	**2.00e+01±0.00e+00**↓	2.04e+01±5.53e-02
f_4	3.44e-05±2.24e-05	3.44e-05±2.24e-05	3.44e-05±2.24e-05	3.44e-05±2.24e-05	9.67e+10±9.13e+10↑	2.50e+09±1.74e+08↑
f_5	2.44e+06±5.19e+05	2.44e+06±5.19e+05	2.44e+06±5.19e+05	2.44e+06±5.19e+05	2.32e+06±4.48e+05	**1.63e+06±3.30e+05**↓
f_6	9.96e+05±4.75e+01	9.96e+05±9.54e+01	9.96e+05±7.24e+01↑	9.96e+05±9.54e+01	9.96e+05±9.83e+01	9.97e+05±8.91e+02↑
f_7	1.87e-08±2.51e-08	1.87e-08±2.51e-08	7.90e+04±7.73e+03↑	6.56e-02±4.86e-02↑	5.87e+07±5.06e+07↑	9.18e+05±6.41e+04↑
f_8	6.17e+03±1.05e+04	**3.60e+03±1.40e+03**↓	1.03e+05±5.11e+04↑	4.65e+03±1.49e+03↓	2.67e+15±3.80e+15↑	2.72e+13±8.04e+12↑
f_9	1.60e+08±3.66e+07	1.60e+08±3.66e+07	1.60e+08±3.66e+07	1.60e+08±3.66e+07	1.60e+08±3.66e+07	2.00e+08±2.26e+07↑
f_{10}	9.08e+07±3.68e+05	9.06e+07±7.13e+04↓	9.05e+07±1.10e+04↓	9.07e+07±7.79e+05↓	9.05e+07±2.23e+05↓	9.08e+07±6.35e+04
f_{11}	**1.31e-08±2.30e-08**	1.50e-08±2.55e-08↑	1.64e+07±1.58e+06↑	5.80e-05±6.48e-05↑	4.43e+10±5.20e+10↑	1.61e+07±1.29e+06↑
f_{12}	9.86e+02±7.61e+01	9.89e+02±7.17e+01↑	9.89e+02±7.17e+01↑	1.01e+03±5.04e+01↑	1.99e+05±3.78e+05↑	1.01e+03±3.20e+01
f_{13}	4.71e+05±5.45e+04	4.79e+05±6.14e+04	4.76e+05±6.12e+04	2.26e+06±2.42e+05↑	4.75e+05±6.10e+04	1.44e+06±1.36e+05↑
f_{14}	2.66e+07±1.90e+06	2.68e+07±1.90e+06↑	2.66e+07±1.90e+06↑	3.44e+07±2.99e+06↑	3.36e+09±4.00e+09↑	2.70e+07±2.05e+06
f_{15}	2.20e+06±2.33e+05	2.20e+06±2.33e+05↑	2.20e+06±2.33e+05↑	2.90e+06±2.79e+05↑	2.20e+06±2.33e+05↓	2.10e+06±2.19e+05
$b/n/l$	—	5/8/2	8/6/1	7/6/2	6/6/3	9/5/1

The symbols ↑ and ↓ denote that the CCFR-ERDG-CMA-ES algorithm performs significantly better than and worse than this algorithm by the Wilcoxon test at the significance level of 0.05, respectively.
Due to the limited space, we show the results to the second decimal place. Some results are shown as the same, but the results actually differ on the value.

TABLE VII

AVERAGE FITNESS VALUES ± STANDARD DEVIATIONS ON THE CEC'2013 FUNCTIONS OVER 25 INDEPENDENT RUNS. THE SIGNIFICANTLY BEST RESULTS ARE IN BOLD FONT (WILCOXON TEST WITH HOLM p-VALUE CORRECTION, $\alpha = 0.05$). b, n, AND l HAVE SIMILAR MEANINGS AS IN TABLE VI

F	CCFR-ERDG -CMA-ES	MMO-CC	CSO	SHADE-ILS	MOS-CEC2013	MA-SW-Chains
f_1	6.07e-17±5.34e-18	4.83e-20±9.45e-21↓	3.63e-17±1.88e-18↓	**2.69e-24±1.35e-23**↓	1.27e-22±7.56e-23↓	8.49e-13±1.11e-12↑
f_2	**4.58e+02±3.68e+01**	1.53e+03±7.42e+01↑	7.08e+02±3.42e+01↑	1.00e+03±8.90e+01↑	8.32e+02±4.57e+01↑	1.22e+03±1.16e+02↑
f_3	2.04e+01±5.45e-02	2.01e+01±1.31e-02↓	2.16e+01±4.09e-03↑	2.01e+01±1.12e-02↓	**9.18e-13±5.23e-14**↓	2.14e+01±5.73e-02↑
f_4	**3.44e-05±2.24e-05**	2.96e+11±3.70e+11↑	1.43e+10±2.23e+09↑	1.48e+08±8.72e+07↑	1.74e+08±8.03e+07↑	4.58e+09±2.51e+09↑
f_5	2.44e+06±5.19e+05	2.80e+06±1.70e+06	**5.67e+05±6.98e+04**↓	1.39e+06±2.03e+05↓	6.94e+06±9.03e+05↑	1.87e+06±3.13e+05↓
f_6	9.96e+05±4.75e+01	1.06e+06±3.21e+03↑	1.06e+06±1.07e+03↑	1.02e+06±1.19e+04↑	**1.48e+05±6.56e+04**↓	1.01e+06±1.56e+04↑
f_7	**1.87e-08±2.51e-08**	1.44e+10±1.27e+10↑	5.51e+06±2.44e+06↑	7.41e+01±5.46e+01↑	1.62e+04±9.29e+03↑	3.45e+06±1.29e+06↑
f_8	**6.17e+03±1.05e+04**	1.77e+14±1.34e+14↑	2.56e+14±7.07e+13↑	3.17e+11±3.06e+11↑	8.00e+12±3.14e+12↑	4.85e+13±1.04e+13↑
f_9	1.60e+08±3.66e+07	1.66e+08±2.94e+07	**3.39e+07±6.90e+06**↓	1.64e+08±1.57e+07	3.83e+08±6.42e+07↑	1.07e+08±1.71e+07↓
f_{10}	9.08e+07±3.68e+05	9.39e+07±7.14e+05↑	9.41e+07±1.98e+05↑	9.18e+07±6.93e+05↑	**9.02e+05±5.17e+05**↓	9.18e+07±1.08e+06↑
f_{11}	**1.31e-08±2.30e-08**	2.72e+12±2.47e+12↑	2.46e+09±2.29e+09↑	5.11e+05±2.25e+05↑	5.22e+07±2.10e+07↑	2.19e+08±3.04e+07↑
f_{12}	9.86e+02±7.61e+01	8.98e+10±2.60e+11↑	1.06e+03±3.39e+01↑	**6.18e+01±1.04e+02**↓	2.47e+02±2.59e+02↓	1.25e+03±1.07e+02↑
f_{13}	4.71e+05±5.45e+04	1.76e+12±1.44e+12↑	4.82e+08±2.12e+08↑	**1.00e+05±7.19e+04**↓	3.40e+06±1.08e+06↑	1.98e+07±1.86e+06↑
f_{14}	2.66e+07±1.90e+06	3.54e+11±4.77e+11↑	1.54e+08±1.26e+08↑	**5.76e+06±3.76e+05**↓	2.56e+07±8.11e+06	1.36e+08±2.15e+07↑
f_{15}	2.20e+06±2.33e+05	4.31e+08±2.06e+08↑	7.62e+07±6.49e+06↑	**6.25e+05±2.40e+05**↓	2.35e+06±1.98e+05↑	5.71e+06±7.73e+05↑
$b/n/l$	—	11/2/2	12/0/3	7/1/7	9/1/5	13/0/2

The symbols ↑ and ↓ have similar meanings as in Table VI.

To show CCFR with ERDG is a competitive solver for large-scale optimization problems, CCFR with ERDG is also compared with MMO-CC [28], CSO [29], SHADE-ILS [30], MOS-CEC2013 [31], and MA-SW-Chains [32]. MMO-CC is a multimodal optimization enhanced CC algorithm. CSO is a competitive swarm optimizer for large-scale optimization. SHADE-ILS, MOS-CEC2013, and MA-SW-Chains were ranked the first in the IEEE CEC'2018, CEC'2013, and CEC'2010 competitions on large-scale global optimization, respectively. Table VII summarizes the results of these algorithms. CCFR–ERDG–CMA-ES significantly outperforms its competitors except SHADE-ILS on most of the CEC'2013 functions. CCFR–ERDG–CMA-ES performs significantly better than SHADE-ILS on seven functions, while performs significantly worse on seven functions. CCFR–ERDG–CMA-ES outperforms its competitors on most of the partially separable functions f_4–f_{11} by several orders of magnitude. CCFR–ERDG–CMA-ES is a competitive algorithm for optimizing the CEC'2013 functions.

V. CONCLUSION

During the binary search of RDG [19] and RDG2 [21], we discovered that the historical information on examining the interrelationship can be used for further interrelationship examination. Based on this discovery, we proved the association between the interrelationship examinations and thus presented an efficient RDG named ERDG for decomposing large-scale continuous problems. By exploiting the historical information on examining the interrelationship between variables, ERDG can avoid several interrelationship examinations, which can reduce the computational cost on decomposing a problem.

ERDG was tested on the IEEE CEC'2010 and CEC'2013 large-scale functions. ERDG spends much fewer FEs decomposing the functions than other peer grouping methods. As the dimensionality of a function increases, ERDG can save more FEs. The saved FEs can be used for optimizing the functions. CCFR, an efficient CC, with ERDG outperforms CCFR with other peer grouping methods on optimizing the CEC'2013 functions. CCFR with ERDG is a competitive solver for the CEC'2013 optimization problems.

In the future, we are planning to investigate the grouping accuracies of the RDG methods including ERDG in decomposing different kinds of large-scale optimization problem and the potential of using reinforcement learning [33] to improve the grouping accuracies of the RDG methods.

SOFTWARE IMPLEMENTATION

The MATLAB source code of the ERDG algorithm can be accessed from the following link: https://github.com/ymzhongzhong/ERDG.

REFERENCES

[1] P. Benner, "Solving large-scale control problems," *IEEE Control Syst. Mag.*, vol. 24, no. 1, pp. 44–59, Feb. 2004.

[2] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Struct. Multidiscipl. Optim.*, vol. 41, no. 2, pp. 219–241, Mar. 2010.

[3] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Proc. IEEE Congr. Evol. Comput.*, 2001, pp. 1101–1108.

[4] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Inf. Sci.*, vol. 316, pp. 419–436, Sep. 2015.

[5] T. Weise, R. Chiong, and K. Tang, "Evolutionary optimization: Pitfalls and booby traps," *J. Comput. Sci. Technol.*, vol. 27, no. 5, pp. 907–936, Sep. 2012.

[6] A. LaTorre, S. Muelas, and J.-M. Peña, "A comprehensive comparison of large scale global optimizers," *Inf. Sci.*, vol. 316, pp. 517–549, Sep. 2015.

[7] M. A. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving From Nature*. Heidelberg, Germany: Springer, 1994, pp. 249–257.

[8] P. Yang, K. Tang, and X. Yao, "A parallel divide-and-conquer-based evolutionary algorithm for large-scale optimization," *IEEE Access*, vol. 7, pp. 163105–163118, 2019.

[9] J. Blanchard, C. Beauthier, and T. Carletti, "A surrogate-assisted cooperative co-evolutionary algorithm using recursive differential grouping as decomposition strategy," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 689–696.

[10] X. Peng, K. Liu, and Y. Jin, "A dynamic optimization approach to the design of cooperative co-evolutionary algorithms," *Knowl. Based Syst.*, vol. 109, pp. 174–186, Oct. 2016.

[11] W. Chen and K. Tang, "Impact of problem decomposition on cooperative coevolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 733–740.

[12] H. Liu, Y. Wang, X. Liu, and S. Guan, "Empirical study of effect of grouping strategies for large scale optimization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3433–3439.

[13] X. Peng and Y. Wu, "Large-scale cooperative co-evolution using Niching-based multi-modal optimization and adaptive fast clustering," *Swarm Evol. Comput.*, vol. 35, pp. 65–77, Aug. 2017.

[14] M. Yang, A. Zhou, C. Li, J. Guan, and X. Yan, "CCFR2: A more efficient cooperative co-evolutionary framework for large-scale global optimization," *Inf. Sci.*, vol. 512, pp. 64–79, Feb. 2020.

[15] D. Yazdani, M. N. Omidvar, J. Branke, T. T. Nguyen, and X. Yao, "Scaling up dynamic optimization problems: A divide-and-conquer approach," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 1–15, Feb. 2020.

[16] Y. Wu, X. Peng, and D. Xu, "Identifying variables interaction for black-box continuous optimization with mutual information of multiple local optima," in *Proc. IEEE Symp. Series Comput. Intell. (SSCI)*, 2019, pp. 2683–2689.

[17] 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, no. 3, pp. 378–393, Jun. 2014.

[18] M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, "DG2: A faster and more accurate differential grouping for large-scale black-box optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 929–942, Dec. 2017.

[19] Y. Sun, M. Kirley, and S. K. Halgamuge, "A recursive decomposition method for large scale continuous optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 647–661, Oct. 2018.

[20] Y. Sun, X. Li, A. Ernst, and M. N. Omidvar, "Decomposition for large-scale optimization problems with overlapping components," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 326–333.

[21] Y. Sun, M. N. Omidvar, M. Kirley, and X. Li, "Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition," in *Proc. ACM Genet. Evol. Comput. Conf. (GECCO)*, 2018, pp. 889–896.

[22] X. Hu, F. He, W. Chen, and J. Zhang, "Cooperation coevolution with fast interdependency identification for large scale optimization," *Inf. Sci.*, vol. 381, pp. 142–160, Mar. 2017.

[23] Y. Mei, M. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Trans. Math. Softw.*, vol. 42, no. 2, pp. 1–24, 2016.

[24] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise. (2010). *Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization*. [Online]. Available: http://goanna.cs.rmit.edu.au/ xiaodong/publications/lsgo-cec10.pdf

[25] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin. (2013). *Benchmark Functions for the CEC'2013 Special Session and Competition on Large Scale Global Optimization*. [Online]. Available: http://goanna.cs.rmit.edu.au/ xiaodong/cec13-lsgo/competition/cec2013-lsgo-benchmark-tech-report.pdf

[26] M. Yang *et al.*, "Efficient resource allocation in cooperative co-evolution for large-scale global optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 493–505, Aug. 2017.

[27] N. Hansen. (2005). *The CMA Evolution Strategy: A Tutorial*. Accessed: 2016. [Online]. Available: https://hal.inria.fr/hal-01297037

[28] X. Peng, Y. Jin, and H. Wang, "Multimodal optimization enhanced cooperative coevolution for large-scale optimization," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3507–3520, Sep. 2019.

[29] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.

[30] D. Molina, A. LaTorre, and F. Herrera, "SHADE with iterative local search for large-scale global optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

[31] A. LaTorre, S. Muelas, and J.-M. Pena, "Large scale global optimization: Experimental results with MOS-based hybrid algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 2742–2749.

[32] D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 1–8.

[33] J. Schmidhuber, "A general method for multi-agent reinforcement learning in unrestricted environments," in *Evolutionary Computation: Theory and Applications*, X. Yao, Ed. Boston, MA, USA: World Sci., 1999, pp. 81–123.

Ming Yang received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the China University of Geosciences, Wuhan, China, in 2005, 2008, and 2012, respectively.

He was a Postdoctoral Researcher with the School of Computer Science, University of Birmingham, Birmingham, U.K., from December 2014 to December 2015. He is an Associate Professor with the School of Computer Science, China University of Geosciences. He also works with the Hubei Key Laboratory of Intelligent Geo-Information Processing, Wuhan. His research interests include swarm intelligence, large-scale optimization, multiobjective optimization, and their applications.

Aimin Zhou (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, Wuhan, China, in 2001 and 2003, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2009.

He is currently a Professor with the School of Computer Science and Technology, East China Normal University, Shanghai, China. He has authored over 70 peer-reviewed papers. His current research interests include evolutionary computation and optimization, machine learning, image processing, and their applications.

Dr. Zhou is an Associate Editor of *Swarm and Evolutionary Computation* and an Editorial Board Member of *Complex and Intelligent Systems*.

Changhe Li (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from the China University of Geosciences, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from the University of Leicester, Leicester, U.K., in July 2011.

He has been an Associate Professor with the China University of Geosciences, Wuhan, since 2012. He is the Vice Chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments. His research interests are evolutionary algorithms with machine leaning, swarm intelligence, multimodal optimization, and dynamic optimization.

Xin Yao (Fellow, IEEE) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technologies, Beijing, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is a Chair Professor of computer science with the Southern University of Science and Technology, Shenzhen, China, and a part-time Professor of computer science with the University of Birmingham, Birmingham, U.K. His current research interests include evolutionary computation, ensemble learning, and their applications to software engineering.

Dr. Yao was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award for his paper on evolving artificial neural networks, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Awards in 2010, 2016, and 2017, the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the Prestigious Royal Society Wolfson Research Merit Award in 2012, the IEEE Computational Intelligence Society (CIS) Evolutionary Computation Pioneer Award in 2013, and many other best paper awards. From 2003 to 2008, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He was the President of IEEE CIS from 2014 to 2015. He was a Distinguished Lecturer of CIS.