Schematic Study on Interaction and Imbalance Effects of Variables for Large-scale Optimization

Sedigheh Mahdavi
Department of Electrical, Computer and Software Engineering
University of Ontario Institute of Technology (UOIT)
Oshawa, Canada
sedigheh.mahdavi@uoit.ca

Shahryar Rahnamayan
Department of Electrical, Computer and Software Engineering
University of Ontario Institute of Technology (UOIT)
Oshawa, Canada
shahryar.rahnamayan@uoit.ca

***Abstract*—In the recent years, Large-Scale Global Optimization (LSGO) algorithms attempt to solve real-world problems efficiently. The imbalance in the contribution of variables and the interaction among variables pose major challenges for LSGO algorithms. This paper proposes mapping schemes based on the interaction among variables and the imbalance in the contribution of variables. The proposed mapping schemes present the different relations between the constructed class of variables according to the interaction feature and the constructed class of variables according to the imbalance feature. Covering a wide range of real-world problems is considered in the mapping schemes; therefore it can provide some insights to design LSGO benchmark suites. By developing LSGO benchmark suites with the ability of representing many-real world problems, researchers will be motivated to realize the success or failure level of LSGO algorithms for tackling various types of LSGO problems. Also, a preliminary set of experiments is conducted to present the importance of considered features in each scheme.**

I. Introduction

Many science and engineering applications include a large number of decision variables, known as Large-Scale Global Optimization (LSGO) problems, such as, designing large-scale electronic systems, scheduling problems with the large number of resources, vehicle routing in large-scale traffic networks, gene recognition in bioinformatics, inverse problem chemical kinetics, etc. Recently, a large number of the metaheuristic algorithms have been proposed to handle the LSGO problems. The canonical metaheuristic algorithms for solving LSGO problems suffer from main deficiency, curse of dimensionality. The major challenges for the performance deterioration of these algorithms are: firstly, the feasible search space grows exponentially with the increase in dimensionality; secondly, increasing size of the problem dimension increases its landscape complexity and characteristic alteration. The LSGO algorithms are classified into two main categories: namely, Cooperative Coevolution (CC) algorithms with the problem decomposition strategy, and non-decomposition based methods. The CC-based approaches are a promising approach to solve LSGO problems [2], [24], [25], [26], [27]. The CC algorithms decompose the LSGO problems into several smaller subcomponents using a decomposition method, and each subcomponent of variables is optimized by a certain optimizer.

Other major challenges which pose complicated difficulties for LSGO algorithms are: the interaction among variables (i.e., level of non-separability) and the imbalance in the contribution of variables on the fitness value [1], [2], [3]. Two variables are non-separable if they cannot be optimized independently to find the global optimum. Recently, most of LSGO algorithms have been developed to solve non-separable LSGO problems. Recently, an imbalance feature was proposed in [1], [3] to develop large-scale optimization benchmark suites for better approximating real-world problems, i.e., problems have some imbalance effects among different subcomponents. Only a limited number of research works have been proposed to solve the imbalance LSGO problems. A Contribution Based Cooperative Co-evolution (CBCC) method was proposed by Omidvar et al. [2], [4], [5]. In the CBCC method, the subcomponent with the maximum effect on the global fitness value is selected for further optimization. Two versions of CBCC method, CBCC1 and CBCC2, are proposed where CBCC1 optimizes the selected subcomponent for only one iteration; while CBCC2 optimizes it until the fitness value is improved. Mahdavi et al. [6] proposed a multilevel optimization framework based on variables effect (MOFBVE). In MOFBVE, first variables with the most influence on the objective function are identified and optimized in the different levels while the values of unimportant variables, i.e., variables with less effect are fixed. MOFBVE contains the several levels with a low-dimension search space of the most influence variables on the fitness value to obtain fitter initial sub-solutions for the original search space as the last level.

In [3], some crucial design features of the CEC-2013 LSGO benchmark test functions were described in more details to provide some guidelines for the design of LSGO benchmark test functions. Although research works in [1], [3] have been developed to the design of LSGO benchmark test functions, they do not pay enough attention to consider the relations between the interaction among variables and the imbalance in the contribution of variables. In this paper, we introduce the new mapping schemes to study the relations among variables which are constructed according to two major features, the interaction among variables and the imbalance in the contribution of variables, in the LSGO problems. Although some studies investigated the aspects of two features, they are not sufficient to cover all kinds of LSGO problems comprehensively. Mapping schemes can better resemble a wider range of LSGO real-world problems and introduce some guidelines to develop benchmark suites for LSGO problems. Also, a preliminary comparative experiments are conducted to demonstrate a general overview of these features on each mapping scheme.

The organization of the rest of this paper is as follows.

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

Section 2 gives a brief background review of LSGO algorithms. Section 3 describes proposed schemes in detail. Section 4 presents a simple algorithm and the experimental results to support discussion on the proposed schemes. Finally, the paper is concluded in Section 5.

II. Background Review

Generally, LSGO algorithms can be divided into two main categorizes, namely, Cooperative Coevolution (CC) algorithms with problem decomposition strategy [7], [8], and non-decomposition based methods [9]. In non-decomposition-based methods, the specific effective operators are developed to enhance the metaheuristic algorithms. Over the past decades, various metaheuristic optimization algorithms including Particle Swarm Optimization (PSO) [10], [11], [12], Evolutionary Algorithms (EAs)[13], [14], Differential Evolution (DE) [15], [16], [17], [18], [19], [20], and Tabu search algorithm [21] have been proposed with focus on the especial alteration such as defining new mutation, selection, and crossover operators, designing and using local search, opposition-based learning [22], [18], sampling operator, hybridization, and incremental or reduction population size methods. The CC algorithms are based on divide-and-conquer approach which decomposes LSGO problems into several low dimensional subcomponents. The main steps of CC framework are as follows: Step 1: Problem decomposition: Dividing an LSGO into some smaller sub-components, Step 2: Subcomponent optimization: Executing individually a traditional optimization algorithm to evolve each subcomponent for predefined iterations in a round-robin method, Step 3: Cooperative combination: Merging the solutions of all subcomponents to construct the n-dimensional solution to evaluate the individuals in each of the subcomponents.

A variety of metaheuristic optimization algorithms have been incorporated into the CC framework for tackling LSGO problems such as evolutionary programming [23], Particle Swarm Optimization (PSO) [24], [25], [26], [27], Artificial Bee Colony (ABC)[28], Evolutionary Algorithms (EAs) [23], [29], and Differential Evolution (DE) [30], [31]. In addition, several large-scale benchmark functions were developed to compare LSGO algorithms. In [1], the CEC-2013 LSGO benchmark functions were introduced with new transformations such as ill-conditioning, symmetry breaking, irregularities, and having subcomponents with non-uniform subcomponent sizes. The standard CC algorithms with the round-robin method divide computational resources equally among subcomponents. The performance of these algorithms deteriorates on the imbalance LSGO problems because by using the round-robin method, a number of the computational budget may be wasted by some subcomponents with a little contribution in the overall fitness value. The CBCC methods attempt to assign more computational budget to the subcomponent with the maximum contribution. For more information on LSGO algorithms, the reader is referred to [9]; a survey paper which has been published recently, covering the CC and non-decomposition based methods to solve LSGO problems.

III. Proposed Mapping Schemes Based on Interaction and Contribution Variable Features

The LSGO problems have two major challenging features: the interaction among variables and the imbalance effect of variables. All decision variables according to the interactions among variables are divided into two general classes: separable and non-separable. There are three general classes of problems according to the variable interaction feature: fully-separable, partially-separable, and fully-non-separable functions [1], [32]. In fully-separable, there is no interaction between any pair of variables. A general form of the partially-separable problems according to the interactions of variables is defined as following formula:

$$F(\vec{x}) = \sum_{i=1}^{m-1} f_i^{nonsep}(x_i) + f_m^{sep}(x_m), \quad x_i = [x_{I_i(1)}, \ldots, x_{I_i(n_i)}]$$

where x_i is mutually exclusive decision vectors of f_i, $X = x_1, \ldots, x_D$ is a global decision vector of D dimensions, and $m(> 1)$ is the number of independent subcomponents. Also, I_i is the index subset of decision variables corresponding to subcomponent i and n_i is the number of variables in the subcomponent i. f_i^{nonsep} and f_m^{sep} indicate the non-separable and separable subcomponents.

Several non-separable subcomponents of interacting variables with no interaction among subcomponents and a separable subcomponent including all separable variables are constructed. It should be noted that a partially-separable problem may include only a set of non-separable subcomponents and no separable sub-component. Furthermore, all decision variables are divided based on their contributions on the fitness value into two classes: significant and non-significant. A general form of the problems according to the imbalance in the contribution of variables and the interaction among variables together is defined as following formula:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i.f_i^{nonsep}(x_i) + f_m^{sep}(x_m), \quad x_i = [x_{I_i(1)}, \ldots, x_{I_i(n_i)}]$$

where w_i is a generated weight to create the imbalance effect. In a problem with m subcomponents, we can sort all subcomponents $(s_1 \ldots s_m)$ according to their effect on the fitness value as follows:

$$s_1 > s_2 > \cdots > s_m \text{ s.t } w_1 > w_2 > \cdots > w_m$$

The significant class includes subcomponents with the most influence on the objective function. Based on the descending arrangement of subcomponents, the significant class consists of the first k subcomponents with the first k maximum weights and all other subcomponents belong to the non-significant class. The parameter k is arbitrary, for instance if we set k to 1, then the significant class includes only the subcomponent with the maximum effect and all other subcomponents are placed in the non-significant class. The value of the parameter k depends on the importance using the information of variables' effect in the LSGO algorithms. In this section, we introduce new mapping schemes to analyze relationships between two categories of variables and their overlap based on two major features, the interaction among variables and the imbalance in the contribution of variables. These mapping schemes demonstrate the relationships of different classes to better resemble a wider range of LSGO real-world problems. Moreover, each mapping scheme has the different importance to represent the real-world problems; it is hard to find only one scheme for all LSGO real-world problems. The most flexible way to create the imbalance among subcomponents is to weight each subcomponent differently [3] therefore in this way, the formal descriptions of mapping schemes are defined by assigning weights for sub-components. In a black-box problem, the recent CC algorithms can identify interaction among variables and construct the non-separable subcomponents and separable subcomponent. Furthermore, by using sensitivity analysis methods [33], [34], the effect of variables in a black-box problem can be computed. By identifying interaction among variables and the imbalance in the contribution of variables, we can recognize which scheme follows a black-box problem.

A. Scheme 1

This mapping scheme illustrates a particular case of problems that all non-separable subcomponents are in the significant class and the separable subcomponent is in the non-significant class. As mentioned above, the significant class consists of the first k subcomponents with the first k maximum weights. Therefore, the weights of non-separable subcomponents must be a value between the minimum and maximum weights of the subcomponents in the significant set which is constructed based on the value k. Also, the separable subcomponent is in the non-significant class therefore the weight of separable subcomponent less than all weights of non-separable subcomponents. A formal description of the mapping scheme 1 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i.f_i^{nonsep}(x_i) + w_m.f_m^{sep}(x_m)\, w_m < w_i \quad (1)$$

Fig. 1 shows a schematic of scheme 1 which represents the relation and overlap among the classes.

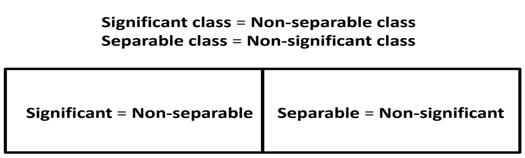

Fig. 1: Scheme 1

B. Scheme 2

Under this mapping scheme, the separable subcomponent belongs to the significant class and all non-separable subcomponents belong to the non-significant class. Therefore, the weights of the separable subcomponent must be greater than the weights of all non-separable subcomponents. A formal description of the scheme 2 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i.f_i^{nonsep}(x_i) + w_m.f_m^{sep}(x_m), \quad (2)$$

$$(\forall i, i = 1, \ldots, m-1), w_m > w_i$$

Fig. 2 shows a schematic of scheme 2 which represents the relation and overlap among the classes.

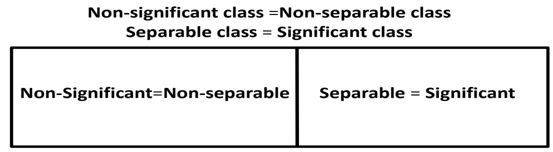

Fig. 2: Scheme 2

C. Scheme 3

In this mapping scheme, the separable subcomponent has some variables in the significant class and some variables in the non-significant class as well. Moreover, all the non-separable subcomponents belong to non-significant class. A formal description of the mapping scheme 3 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i.f_i^{nonsep}(x_i) + \quad (3)$$

$$w_m.f_m^{sep}(x_m) + w_{m+1}.f_m^{sep}(x_{m+1}) \quad (4)$$

$$(\forall i, i = 1, \ldots, m), w_{m+1} > w_i$$

The mapping scheme assumes that the value k is set to 1 therefore the significant class includes only the separable subcomponent with the maximum effect. Fig. 3 shows a schematic of scheme 3 which represents the relation and overlap among the classes.

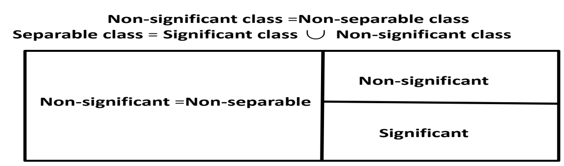

Fig. 3: Scheme 3

D. Scheme 4

The mapping scheme 4 demonstrates a case of problems that the non-separable subcomponents have the different weights to generate the imbalance such that some of the non-separable subcomponents are belonging to the non-significant and other non-separable subcomponents are in the significant class. The separable subcomponent is in the non-significant class. A formal description of the mapping scheme 4 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i.f_i^{nonsep}(x_i) + w_m.f_m^{sep}(x_m) \quad (5)$$

$$(\forall i, i = 1, \ldots, k), (\forall j, j = k+1, \ldots, m), w_i > w_j$$

Where k is the number of significant subcomponents in the significant class as mentioned above. Fig. 4 shows a schematic of scheme 4 which represents the relation and overlap among the classes.

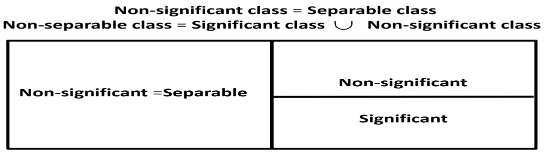

Fig. 4: Scheme 4

E. Scheme 5

Under this mapping scheme, the significant class contains some non-separable subcomponents and the separable subcomponent. Also, the other non-separable subcomponents belong to the non-significant class. A formal description of the mapping scheme 5 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{k} w_i . f_i^{nonsep}(x_i) + \sum_{i=k+1}^{m-1} w_i . f_i^{nonsep}(x_i) \quad (6)$$

$$+ w_m . f_m^{sep}(x_m) \; (\forall i, i = 1, \ldots, k \,\&\, i = m), \quad (7)$$

$$(\forall j, j = k+1, \ldots, m-1), \; w_i > w_j$$

Where k is the number of significant subcomponents in the significant class as mentioned above. Fig. 5 shows a schematic of scheme 5 which represents the relation and overlap among the classes.

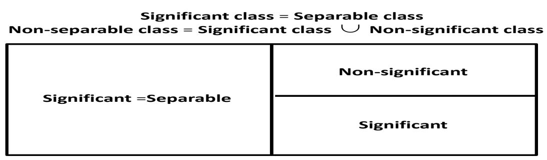

Fig. 5: Scheme 5

F. Scheme 6

In this mapping scheme, the significant class includes all non-separable subcomponents and some separable variables as well. The other separable variables are the members of the non-significant class. A formal description of the mapping scheme 6 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m-1} w_i . f_i^{nonsep}(x_i) + w_m . f_m^{sep}(x_m) + \quad (8)$$

$$w_{m+1} . f_{m+1}^{sep}(x_{m+1}) \quad (9)$$

$$(\forall i, \, i = 1, \ldots, m-1 \,\&\, i = m), \; w_i > w_{m+1}$$

Fig. 6 shows a schematic of scheme 6 which represents the relation and overlap among the classes.

G. Scheme 7

Both non-significant and significant classes have the elements of non-separable and separable subcomponents. A

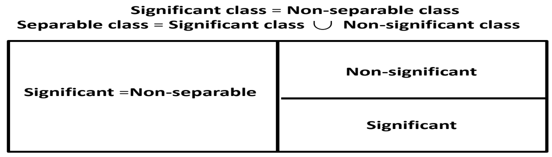

Fig. 6: Scheme 6

formal description of the mapping scheme 7 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{k} w_i . f_i^{nonsep}(x_i) + \sum_{i=k+1}^{m-1} w_i . f_i^{nonsep}(x_i) \quad (10)$$

$$+ w_m . f_m^{sep}(x_m) \; (\forall i, i = 1, \ldots, k \,\&\, i = m), \quad (11)$$

$$(\forall j, j = k+1, \ldots, m-1 \,\&\, j = m+1), \; w_i > w_j$$

Fig. 7 shows a schematic of scheme 7 which represents the relation and overlap among the classes.

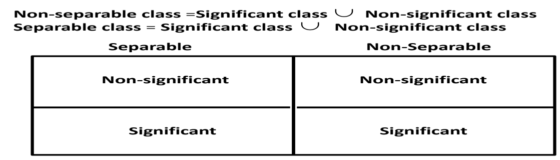

Fig. 7: Scheme 7

H. Scheme 8

This mapping scheme contains partially-separable problems with only a set of non-separable subcomponents and no fully-separable subcomponent. In the scheme, some non-separable subcomponents are inside significant class and other significant class inside non-significant class. A formal description of the mapping scheme 8 is defined as follows:

$$F(\vec{x}) = \sum_{i=1}^{m} w_i . f_i^{nonsep}(x_i) \quad (12)$$

Fig. 8 shows a schematic of scheme 8 which represents the relation and overlap among the classes

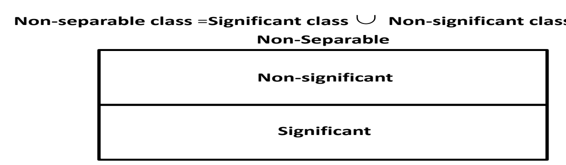

Fig. 8: Scheme 8

I. Scheme 9

This mapping scheme contains separable problems. Some separable variables are inside significant class and other separable variables inside non-significant class. Generally, each two non-significant and significant classes have one subcomponent therefore the weight of subcomponent in the significant class must be greater than the weights of subcomponent in the non-significant. A formal description of the mapping scheme 9 is defined as follows:

$$F(\vec{x}) = w_1.f_1^{sep}(x_1) + w_2.f_2^{sep}(x_2)\, w_2 > w_1 \qquad (13)$$

Fig. 9 shows a schematic of scheme 9 which represents the relation and overlap among the classes.

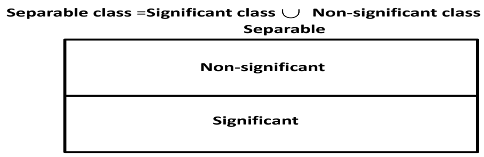

Fig. 9: Scheme 9

IV. Experiments

This section first used a simple algorithm similar to MOFBVE [6] to consider imbalance feature on the LSGO algorithms, then reports the obtained results on all mapping schemes. The results reported in this study give a general overview of considering imbalance and interaction variables by proposing mapping schemes as a platform to systematically study these features of variables therefore much more effort is needed to investigate further the analysis of mapping schemes and optimal algorithms; which is beyond the scope of this paper. In order to analyze the impact of using imbalance feature on the LSGO algorithms, we implemented a simple method (CC-ideal1); the CC algorithm with the ideal decomposition method (CC-ideal). Both CC-ideal1 and CC-ideal employ the ideal decomposition method as decomposition method in their CC framework which the ideal decomposition method constructs subcomponents manually using the knowledge of benchmark functions. In CC-ideal, after building subcomponents, it optimizes subcomponents individually in a round-robin fashion but CC-ideal1 utilizes the imbalance knowledge of a given problem to recognize the significant subcomponent with maximum contribution on the fitness value. The main idea of CC-ideal1 is that, first; the significant subcomponent is identified by the imbalance knowledge of a problem and optimized instead of the whole decision variables at a few iterations before beginning of the optimization process in the CC-ideal algorithm. When this subcomponent is being optimized, all other variables are kept fixed. The best members of initialization populations are employed to the other variables. Then, the obtained solutions from the optimization of the significant subcomponent will be used as the initial population at process in the CC-ideal algorithm. Furthermore, the significant subcomponent is optimized at a few determined iterations ($10D_1$ or $20D_1$, where D_1 is the dimension of the significant subcomponent).

A. *Numerical Results*

In this section, we provided a preliminary series of experiments to indicate a general overview of how on each mapping scheme, using the significant variables can affect on the performance of the CC algorithm with the ideal decomposition method (CC-ideal). We used some partially separable functions which are based on the CEC-2013 LSGO benchmark suit to a preliminary comparison of CC-ideal algorithm with CC-ideal algorithm using the significant subcomponent (called CC-ideal1). For CC-ideal algorithm, 2 (f_4 and f_7) partially separable functions are changed to cover first 7 mapping schemes; respectively. These functions (f_4, f_5, and f_7) have a set of non-separable subcomponents and one fully-separable subcomponent. In scheme 1, for all functions, the weight of all non-separable subcomponents is changed to the maximum weight of subcomponents in the CEC-2013 LSGO benchmark suite. For all functions of scheme 2, the weight of all non-separable subcomponents is changed to 1 and the weight of all non-separable subcomponents is assigned to the fully-separable subcomponent.

In scheme 3, the fully-separable subcomponent is divided into two separable subcomponents with the same size (350) and 1 and 10^6 weights which is greater than the maximum weight of non-separable subcomponents. The partially separable functions in the CEC-2013 LSGO benchmark suite are in scheme 4. For scheme 5, the maximum weight of non-separable subcomponents is assigned to the fully-separable subcomponent. For all functions of scheme 6, the weight of all non-separable subcomponents is changed to the maximum weight of subcomponents in the CEC-2013 LSGO benchmark suite. Then, the fully-separable subcomponent is divided to two separable subcomponents with same size (350) which have the weights 1 and same weight with non-separable subcomponents. In scheme 7, the fully-separable subcomponent is divided to two separable subcomponents with same size (350) which have the weights 1 and the maximum weight of non-separable subcomponents. All functions of scheme 8 (f_8,f_9, and f_{10}) are selected from the CEC-2013 LSGO benchmark suite without any change. Functions of scheme 9 are three separable functions (f_1, f_2, and f_3) in the CEC-2013 LSGO benchmark suite which 200 variables of problems have weight 10^6. Also, a two-sided Wilcoxon statistical test with a confidence interval of 95% is performed for comparison of algorithms; the better results are highlighted in bold-face. In this study, the maximum number of evaluations was set to 3×10^6, the population size was set to 50, and all algorithms were evaluated for 25 independent runs and the results were recorded.

1) Results for scheme 1: Table I shows the results of CC-ideal and CC-ideal1 algorithms. From Table I that in comparison with CC-ideal, CC-ideal1 can obtain better results on all two test functions.

TABLE I: Results of CC-ideal and CC-ideal1 on the scheme 1.

Function		CC-ideal	CC-ideal1
f_4	Mean	4.34e+11	**1.10e+11**
	std	5.12e+10	2.20e+10
f_7	Mean	2.28e+10	**2.00e+09**
	std	5.81e+09	3.88e+09

2) Results for scheme 2: Table II shows the results of CC-ideal and CC-ideal1 algorithms. It can been seen from Tables II that CC-ideal have the worse results compared to CC-ideal1. Based on results, when the significant class contains only all separable variables, the performance of CC-ideal1 is degraded.

3) Results for scheme 3: Table III show the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table III

TABLE II: Results of CC-ideal and CC-ideal1 on the scheme 2.

Function		CC-ideal	CC-ideal1
f_4	Mean	**4.23e+12**	6.51e+12
	std	6.47e+12	5.26e+12
f_7	Mean	**2.99e+09**	5.79e+09
	std	2.16e+09	4.97e+09

that in comparison with CC-ideal, CC-ideal1 can obtain better results on all two test functions. These results indicate that, in the scheme 3, using significant subcomponents can improve the efficiencies of CC-ideal algorithm.

TABLE III: Results of CC-ideal and CC-ideal1 on the scheme 3.

Function		CC-ideal	CC-ideal1
f_4	Mean	4.65e+10	**1.93e+10**
	std	1.63e+10	2.15e+10
f_7	Mean	5.45e+07	**5.03e+06**
	std	1.95e+07	7.91e+06

4) Results for scheme 4: Table IV shows the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table IV that in comparison with CC-ideal, CC-ideal1 can obtain better results on all two test functions. These results indicate that, in the scheme 4, using significant subcomponents can improve the efficiencies of CC-ideal algorithm.

TABLE IV: Results of CC-ideal and CC-ideal1 on the scheme 4.

Function		CC-ideal	CC-ideal1
f_4	Mean	4.97e+10	**4.81e+08**
	std	1.97e+10	1.77e+08
f_7	Mean	6.33e+07	**3.06e+07**
	std	2.36e+07	5.24e+07

5) Results for scheme 5: Table V shows the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table V that the results of CC-ideal are better than or comparable to CC-ideal1 on all two test functions. As mentioned above, once again these results indicate that, in the scheme 5, using significant subcomponents can lead to degrade the performance of CC-ideal1 algorithm when the significant class contains only all separable variables like scheme 2.

TABLE V: Results of CC-ideal and CC-ideal1 on the scheme 5.

Function		CC-ideal	CC-ideal1
f_4	Mean	**9.84e+10**	**1.08e+11**
	std	6.83e+10	6.87e+10
f_7	Mean	**1.70e+08**	4.50e+08
	std	9.82e+07	2.03e+08

6) Results for scheme 6: Table VI shows the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table VI that in comparison with CC-ideal, CC-ideal1 can obtain better results on all two test functions. These results indicate that, in the scheme 6, using significant subcomponents can improve the efficiencies of CC-ideal algorithm. In scheme 1, the significant class contains all non-separable subcomponents while in this scheme some of all non-separable subcomponents are in the significant class. In both schemes 1 and 6, using significant subcomponents can improve the efficiencies of CC-ideal algorithm on these benchmark functions.

TABLE VI: Results of CC-ideal and CC-ideal1 on the scheme 6.

Function		CC-ideal	CC-ideal1
f_4	Mean	4.38e+11	**2.36e+11**
	std	4.96e+10	1.20e+11
f_7	Mean	2.51e+10	**2.63e+09**
	std	4.31e+09	9.13e+08

7) Results for scheme 7: Table VII shows the results of CC-ideal and CC-ideal1 algorithms. From Table VII that in comparison with CC-ideal, CC-ideal1 can obtain better results on all two test functions. Based on results, it is expected that the performance of CC algorithms is enhanced significantly especially when two non-significant and significant classes have the non-separable and separable subcomponents together.

TABLE VII: Results of CC-ideal and CC-ideal1 on the scheme 7.

Function		CC-ideal	CC-ideal1
f_4	Mean	5.09e+10	**4.17e+09**
	std	1.90e+10	2.27e+09
f_7	Mean	6.22e+07	**2.75e+06**
	std	2.45e+07	3.59e+06

8) Results for scheme 8: Table VIII shows the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table VIII that in comparison with CC-ideal, CC-ideal1 can obtain better results on two test functions although CC-ideal outperforms CC-ideal1 on only one function.

TABLE VIII: Results of CC-ideal and CC-ideal1 on the scheme 8.

Function		CC-ideal	CC-ideal1
f_8	Mean	4.86e+15	**4.86e+14**
	std	1.85e+15	1.96e+15
f_9	Mean	4.97e+08	**4.31e+08**
	std	3.53e+07	9.50e+07
f_{10}	Mean	**1.64e+01**	6.41e+01
	std	1.96e+01	2.58e+01

9) Results for scheme 9: Table IX shows the results of CC-ideal and CC-ideal1 algorithms. It is obvious from Table IX that in comparison with CC-ideal1, CC-ideal1 can obtain better results on one test functions.

TABLE IX: Results of CC-ideal and CC-ideal1 on the scheme 9.

Function		CC-ideal	CC-ideal1
f_1	Mean	**2.99e+06**	**2.58e+06**
	std	1.19e+06	3.65e+06
f_2	Mean	**3.50e+08**	**3.50e+08**
	std	5.34e+07	5.35e+07
f_3	Mean	**2.19e+06**	2.57e+06
	std	2.97e+05	3.95e+05

V. Discussions on the LSGO benchmark suit and LSGO algorithms

Based on the proposed schemes, the design of new LSGO benchmark test functions becomes an essential challenge to cover the characteristics of real-world problems. In following, we describe some potential areas of improvement and challenges which motivate interested researchers in LSGO fields to investigate some possible modifications of LSGO benchmark test functions and also describing how LSGO algorithms can be influenced by considering all features in the design of the LSGO benchmark suit.

1) Identifying the significant variables
In all mentioned classes in all mapping schemes, a critical task is identifying the significant variables. A main factor of identifying the significant variable are the size and weight subcomponents. For example, in the function $w_1.f_1^{nonsep}(x_1)+w_2.f_2^{sep}(x_2)$ with 1000 variables including a class of non-separable subcomponent and a class of separable subcomponent. For example if we know $w_1 > w_2$, it will be definitely asked; is the non-separable subcomponent with the greater weight always located in the significant class? Obviously, the answer of this question relies on some factors; (1) the number of non-separable variables against the number of separable variables; for instance if there are only 10 non-separable variables among 1000 variables and all other variables be separable so it is hard to have a certain response for this question. In fact, the size of a subcomponent can play a key role to make this subcomponent as the significant subcomponent which has a higher contribution on the overall objective function. (2) the weight of each subcomponent; for instance in the function $w_1.f_1^{nonsep}(x_1) + w_2.f_2^{nonsep}(x_2)$ with 1000 variables including two classes of non-separable subcomponents, when the weight of a subcomponent is greater than other one (e.g., $w2 > w1$), it can not be concluded that this subcomponent is definitely located in the significant class. The reason for this is that, two main factors should be considered which they can influence on a subcomponent to have higher contribution on the objective function as the significant class; (1) how much is the difference among the weights of subcomponents? (2) what is the number of variables in subcomponents? Further research can be directed to address unanswered questions in both fields; LSGO benchmark test functions and LSGO algorithms. Some of these questions are: How can we identify the significant class by considering the difference among the weights of all subcomponents and also their size? How many variables can be considered as significant variables? How the number of significant variables can affect on the performance of LSGO algorithms? What are differences among the various values of variables' contribution? How can new optimization algorithms be developed based on the amount of these differences? What is the minimum value of differences among contributions to be beneficial for optimization algorithms for considering significant variables? These questions may be directly related to solving the given problem and the optimization algorithm.
2) Impact of significant variables on the computational budget assignment methods
A significant challenge of using significant variables is how the LSGO algorithms can efficiently allocate computational budget based on their contribution. In the non-decomposition based methods, the number of optimization iterations for significant variables poses a challenge on these methods. In new proposed schemes, it is desirable to investigate the optimal budget assignment of significant variables. In [2], [3], [4], a challenge was proposed for the CC algorithms with the ability budget assignment such that they can allocate the computational budget among all subcomponents based on their different contributions on the objective function. Also, it has shown that assigning more budgets to the subcomponent with the maximum contribution has a significant effect on the performance of CC algorithms. In [3], it has shown that a major challenge of contribution-based CC is solving problems with the non-uniform weighting of the subcomponents. CBCC methods consider only the subcomponent with the maximum contribution. Further research can be directed to extend proper budget assignment methods in order to use in CC algorithms. It is desirable to introduce some major aspects of budget assignment methods.
Another key aspect is investigating impact of subcomponents' size. When a budget assignment method is designed based on only the contributions of subcomponents, it may not reflect the effect of the size of sub-components as mentioned before the size of subcomponents can affect on their contribution on the overall objective function. It is mentioned in [3] that non-uniform dimensionality of subcomponent and weighting approach create different contributions of subcomponent; but some cases can be happened that two subcomponents have approximately the same contribution while they have the different dimension sizes. For instance, if a problem has four subcomponents with 50, 70, 100, and 120 dimensions such that the weight of components with the size 50 be two times more than the weight of components with the size 100 so they have the same contribution. Another aspect for budget assignment methods is determining types of mapping schemes which using budget assignment method can improve the performance of LSGO algorithms. Therefore, a great potential future research works can be investigating thoroughly the strengths and weaknesses of budget assignment methods and finding definite reasons for good or poor performance of these methods.

VI. Conclusions

In this paper, variant schemes were introduced according to the interaction among variables and the imbalance in the contribution of variables. These mapping schemes provide a straightforward mapping to present a wide range of real-world problems. According to the theorem of "no free theorems" [35], one algorithm cannot offer better performance than the others in every aspect on all class of problems. Therefore, for designing efficient algorithms we proposed mapping schemes which consider the different type of problems as mapping schemes which would be beneficial for find proper algorithms for each class of problems. Preliminary experiments are conducted on all mapping schemes by modifying some functions of the CEC-2013 LSGO benchmark suite to cover all mapping schemes. The observations indicate that considering imbalance feature by using mapping schemes has potential to play a crucial role in LSGO algorithms. This study is a preliminary investigation and introduction and we have identified the following potential areas based on the proposed mapping schemes of improvement and challenges which motivate interested researchers in LSGO fields to investigate the use of significant variables on these schemes: (1) the design of computational budget assignment strategy for LSGO algorithms to allocate computational budget according to the contribution of variables, (2) the design of new benchmark test functions based on the proposed schemes, (3) investigating the impact of the dimension size of subcomponents on the budget assignment method.

REFERENCES

[1] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the cec 2013 special session and competition on large-scale global optimization," *gene*, vol. 7, no. 33, p. 8, 2013.

[2] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 378–393, June 2014.

[3] M. N. Omidvar, X. Li, and K. Tang, "Designing benchmark problems for large-scale continuous optimization," *Information Sciences*, 2015.

[4] M. N. Omidvar, X. Li, and X. Yao, "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 1115–1122.

[5] B. Kazimipour, M. N. Omidvar, X. Li, and A. Qin, "A sensitivity analysis of contribution-based cooperative co-evolutionary algorithms," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 417–424.

[6] S. Mahdavi, S. Rahnamayan, and M. E. Shiri, "Multilevel framework for large-scale global optimization," *Soft Computing*, pp. 1–30, 2016.

[7] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving from NaturePPSN III*. Springer, 1994, pp. 249–257.

[8] M. A. Potter, "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, Citeseer, 1997.

[9] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Information Sciences*, vol. 295, pp. 407–428, 2015.

[10] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *Cybernetics, IEEE Transactions on*, vol. 45, no. 2, pp. 191–204, 2015.

[11] M. A. M. de Oca, D. Aydın, and T. Stützle, "An incremental particle swarm for large-scale continuous optimization problems: an example of tuning-in-the-loop (re) design of optimization algorithms," *Soft Computing*, vol. 15, no. 11, pp. 2233–2255, 2011.

[12] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.

[13] W. Dong, T. Chen, P. Tino, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 6, pp. 797–822, 2013.

[14] Y. Wang, J. Huang, W. S. Dong, J. C. Yan, C. H. Tian, M. Li, and W. T. Mo, "Two-stage based ensemble optimization framework for large-scale global optimization," *European Journal of Operational Research*, vol. 228, no. 2, pp. 308–320, 2013.

[15] C. García-Martínez, F. J. Rodríguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation," *Soft Computing*, vol. 15, no. 11, pp. 2109–2126, 2011.

[16] A. LaTorre, S. Muelas, and J.-M. Peña, "A mos-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test," *Soft Computing*, vol. 15, no. 11, pp. 2187–2199, 2011.

[17] C. Wang and J.-H. Gao, "A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization," *Optimization Letters*, vol. 8, no. 2, pp. 477–492, 2014.

[18] H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing*, vol. 15, no. 11, pp. 2127–2140, 2011.

[19] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, no. 1, pp. 153–177, 2012.

[20] M. Weber, F. Neri, and V. Tirronen, "Shuffle or update parallel differential evolution for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2089–2107, 2011.

[21] A.-R. Hedar and A. F. Ali, "Tabu search with multi-level neighborhood structures for high dimensional problems," *Applied Intelligence*, vol. 37, no. 2, pp. 189–206, 2012.

[22] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 64–79, 2008.

[23] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2. IEEE, 2001, pp. 1101–1108.

[24] J. Fan, J. Wang, and M. Han, "Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods," *Fuzzy Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 829–839, 2014.

[25] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 2, pp. 210–224, 2012.

[26] L. Sun, S. Yoshida, X. Cheng, and Y. Liang, "A cooperative particle swarm optimizer with statistical variable interdependence learning," *Information Sciences*, vol. 186, no. 1, pp. 20–39, 2012.

[27] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 225–239, 2004.

[28] Y. Ren and Y. Wu, "An efficient algorithm for high-dimensional function optimization," *Soft Computing*, vol. 17, no. 6, pp. 995–1004, 2013.

[29] H. K. Singh and T. Ray, "Divide and conquer in coevolution: A difficult balancing act," in *Agent-Based Evolutionary Search*. Springer, 2010, pp. 117–138.

[30] 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.

[31] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[32] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization," Nature Inspired Computation and Applications Laboratory (NICAL),USTC, China, Tech. Rep., 2010. [Online]. Available:

[33] H. Rabitz and Ö. F. Aliş, "General foundations of high-dimensional model representations," *Journal of Mathematical Chemistry*, vol. 25, no. 2-3, pp. 197–233, 1999.

[34] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[35] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, Evolutionary Computation, IEEE Transactions on 1 (1) (1997) 67–82