Empirical Study of Effect of Grouping Strategies for Large Scale Optimization

Haiyan Liu, Yuping Wang†, Xuyan Liu, Shiwei Guan
School of Computer Science and Technology
Xidian University
Xi'an, Shanxi, 710071, China
Email: hyliu83@126.com; ywang@xidian.edu.cn

Abstract—**The cooperative co-evolution framework (CC) is widely used in the large scale global optimization. It is believed that the CC framework is very sensitive to grouping strategies and the performance deteriorate if interacted variables are not correctly grouped. So many efforts have been devoted to find good ways to correctly decompose the large scale problem into smaller sub-problems so as to effectively solve the original problem by optimizing these smaller sub-problems using a search algorithm. However, what is the relationship between the grouping strategy and the search algorithm adopted in CC? what is the effect of grouping strategies on the CC framework? This work will tackle these issues. We try to unveil the impact of different grouping strategies on CC and the relationship between the grouping strategies and the search algorithms by empirical study. The experiment results show that the correct result of variable grouping is very important since it can turn the large scale problem into smaller sub-problems and make the problem solving easier. It indeed has a big influence on the results obtained by the search algorithm. However, when the search algorithm adopted is not suitable or effective, even if the grouping strategy gives the correct grouping results, the final results may be poor. In this case, grouping strategy only plays little role on the CC. Thus, only effective grouping strategy plus efficient search algorithm can result in good solutions for large global optimization problems.**

I. Introduction

Some of the real world problems are quite complex or contain a large number of decision variables, these problems especially with a dimension no less than 1000 are often referred to as large scale problems. In recent years, large scale global optimization (LSGO for short) has attracted many interests and a lot of efforts have been devoted to it. The traditional optimizing methods such as Newton or quasi-Newton methods usually can't be directly and effectively applied to these problems. Evolutionary algorithms (EA) [1, 2] are promising optimization methods for a very wide range of optimization problems. However, their performances deteriorate rapidly with the increase of the dimensionality [3]. Potter and De Jong adopted the divide-and-conquer strategy and proposed a CC framework [4]. In their first implementation, they decomposed the n dimensional problem into n one dimensional sub-problems. The general framework of CC is as follows. First, the LSGO problem is divided into sub-components (subgroups); this process is called grouping or decomposition. Then each sub-component is evolved separately using a special search algorithm (e.g., an evolutionary algorithm).

It seems that variable interaction will greatly affect the results of the search algorithm. So it is necessary to identify the interacting variables before the optimization process. But only a few works have been done on this issue. The work in [5] showed that the interactions between variables will significantly slow down the convergence of optimization algorithms. But the experiments in [5] are all small scale with dimensions no more than 30. In literature [6], the author compared the random and ideal (correct) grouping strategy using basic DE algorithm on CEC' 2010 benchmark functions [7]. For the 20 benchmark functions with dimension 1000, ideal grouping wins 10, loses 6 and ties 4 versus random grouping. Note that basic DE algorithm is not very powerful compared to some new DE variants, the question is can some better algorithm than basic DE fill the gap between ideal grouping and random grouping? Also,they used the fixed sub-component dimension m=50 and didn't use the better parameter values of DE for large scale problems suggested in literature [8], which may affect the performance and the result may not be precise. Furthermore, they used CEC' 2010 benchmark suit which is much easier than benchmark suit CEC' 2013. It can not be guaranteed that it can act similarly well if the more complicated CEC' 2013 benchmark suit [9] is used.

All these issues need to be further studied. To do so, we carry out a series of experiments on the latest proposed benchmark suit CEC' 2013 of large scale optimization problems [9] and used a more efficient search algorithm named SaNSDE proposed in [10] combining different grouping strategies. First, we conduct random grouping with different group sizes to see if grouping size have any effect on the optimization and choose the best grouping size for random grouping. Second, we make comparison between the best random grouping (with the group size that achieves the best performance of random grouping) and the ideal grouping to get better understanding of the influence of variable interactions. Third, we compare the result of different grouping strategies and no grouping strategy to see the effect of variable grouping. Overall, in this paper we will try to answer the following questions:

1) For a certain algorithm, does it make a difference when grouping strategy is applied?
2) For fully separable or fully non-separable problems, should we still apply some grouping strategies?
3) How much is the influence of correct grouping on the

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

optimization algorithm?
4) Does grouping size affect the optimization results on large scale problems?

We believe that the discussions on the above issues will shed some light on the large scale optimization.

The rest of this paper is organized as follows: Section II presents the background of LSGO and the related techniques. Section III gives the detailed description of the experiments and the results as well as discussions. Section IV is the conclusion and future works.

II. Background

In this section, we gave a brief review of the techniques for large scale optimization problems. Generally speaking, the optimization algorithms for LSGO fall into two categories:
1) With variables decomposition strategy and optimize each subgroup under CC framework;
2) Without explicit decomposition, optimize the problem using hybrid local search algorithms.

First, we introduce some variable decomposition techniques (grouping strategies). It can further be divided into two categories: black box technique and white box technique. In the resent years a lot of variable interaction identification and grouping strategies have been proposed: Yang et al proposed a random grouping strategy under the CC framework called DECC-G (Differential Evolution with Cooperative co-evolution and Random grouping) [11]. DECC-G randomly decomposes the decision vector into m subgroups where m is given by the user, so each subgroup has the same size: $dimention/m$. In literature [12], MLCC (Multilevel Cooperative Co-evolution) was proposed which is similar to DECC-G but the subgroup size is chosen from a set of potential group sizes. This strategy performs better than DECC-G because it self-adaptively changes the component size according to the historical performance. Then Omidvar et al proved that random grouping is ineffective when the number of interacting variables grows more than five [13]. So he proposed a revised random grouping strategy named DECC-ML, which could result in finding a solution using fewer function evaluations [13]. However, it is obviously that when the number of interacting variables get larger, even this revised random grouping is unable to identify and group these variables correctly. In literature [14] DECC-D (DECC with delta grouping) was proposed using delta grouping. The main idea of delta grouping is that when interacting variables are not grouped within the same subgroup (wrong grouping), improving interval (improvement of the function value, delta value) is limited. So they compute the change of the function value (delta value) for each variable and group the variables with small delta values into one group. However when there are more than one group of interacting variables, delta grouping is unable to identify them. Sun et al proposed DECC-CIG (DECC with Correlation Identification Grouping) algorithm [15] that capture interacting variables using the entropy based measure called symmetrical uncertainty. The entropy based measure can only handle discrete values, continuous values should be discreted properly which is not easy. Weicker et al gave an insight of finding variable interaction in literature [16]. Chen et al [17] improved this technique and proposed CCVIL (Cooperative Co-evolution with Variable Interaction Learning) for solving large scale problems. A contribution based cooperative co-evolution algorithm is proposed in literature [18], by reducing the imbalance in the contribution to the global fitness between the separable and non-separable subgroups, the algorithm can gain more efficiency. In literature [19], a new automatic decomposition strategy called differential grouping is proposed. Based on this a differential grouping algorithm with CC framework called DECC-DG was proposed to solve the large scale problems. It is effective and achieve good result on CEC' 2010 bench mark suit. However, the differential grouping in [19] only can identify direct interactions among variables while the indirect interaction is neglected. In literature [20], the extended differential grouping is proposed with the ability of identifying indirect interactions among variables. [21] further improved the differential grouping and adopted a modified CMA-ES (Covariance Matrix Adaptation-Evolutionary Strategy) as the optimizer of sub-problems. For more detail about CMA-ES, please refer to [22].

All of the above mentioned grouping strategies are black box methods. Authors in [23, 24] first proposed a white box grouping strategy in literature in which the function expression (formula) is used for detecting variable interaction. We will refer to this white-box grouping strategy as FBG (Formula based grouping) hereinafter.

Besides the decomposition based methods under CC framework, there are other techniques for solving LSGO problems. MA-SSW-Chains [25] is a memetic algorithm that uses steady-state GA as the global search method and the Subgrouping Solis Wets' algorithm as the local search method. It is the winner of competition on high-dimensional global optimization at WCCI 2010 [26]. Multiple trajectory Search (MTS) [27] uses multiple (three) local search algorithms on a population generated using simulated orthogonal array. SaDE-MMTS (Self-adaptive DE with Modified MTS) in literature [28] corporates SaDE and the modified MTS to solve the LSGO problem. Multiple Offspring Sampling based hybrid algorithm (MOS) [29] builds a framework which seamlessly combines multiple algorithms in a dynamic way. Currently MOS achieves the best results on CEC' 2013 benchmark suit for LSGO and does not use the decomposition.

This may arise the following questions:" Does the LSGO need the decomposition? ", "Does variable interaction really play a very important role on LSGO?", "Why methods without taking account of variable interaction also have very good or even better results?". This paper tries to discuss these issues using solid experiments.

III. Experiments and Results

This study mainly investigates the effects of decomposition and variable interactions on large scale optimization. To achieve this purpose, we use ideal grouping and random grouping as the comparison objects. The ideal grouping means the grouping strategy which can always correctly group all interacting variables, while random grouping represents the grouping strategy which does not consider variable interactions and group variables randomly. To make the investigation more reliable, the experiments are conducted with three strategies: with random grouping, no grouping and ideal grouping. The experiments can be divided into two parts. The first part tries to find if there is a significant difference among different group sizes of random grouping. The second part compares the performance of CC with ideal grouping, random grouping and no grouping.

A. Experimental Set up

The experiments are executed on the CEC' 2013 benchmark suit for large scale optimization [9]. There are 15 test functions in which f1-f3 are fully separable functions, f4-f11 are partially (additively) separable functions and f12-f14 are overlapping functions (non-separable) and f15 is non-separable function. The pseudo-code of the algorithm we used is summarised in algorithm 1, where SaNSDE [10] is one of the most efficient DE variants, and we choose it as the optimizer for all the three grouping strategies. For each

Algorithm 1 pseudo-code of the algorithm used in the experiments

Initialization:
set FE=0 and randomly initialize NP points to form the population.
group $\leftarrow$ different_group_strategies
choose the best point as the parameter for SaNSDE
while $FE < MaxFEs$ **do**
  for $i = 1$ to $group\_num$ **do**
    use SaNSDE to optimize each group under CC framework.
    $FE = FE + usedFE$;
  end for
end while

function and each grouping strategy, we run algorithm 1 for 15 independent runs, where the grouping strategies used are as follows:

1) Ideal grouping: since the white box variable grouping strategy FBG [24] can correctly decompose all the test functions, we use its results as the ideal grouping strategy.
2) Random grouping: Randomly decompose the variables into the same size subgroups. We test 6 cases of decomposition with the size of each subgroup being 10, 30, 50, 100, 150, and 200 respectively. For each case, we conduct 15 independent runs.
3) Random grouping with the same group sizes as ideal grouping: For partially separable functions f4-f11, the random grouping takes the same size subgroups as those of the ideal grouping, but for each group, the variables are randomly chosen.
4) No grouping: No decomposition is made and the all variables will be optimized as a whole.

In the experiments ,the population size is set to 50 and the maximum number of function evaluations (MAXFEs) is set to 3e+6.

B. Results and Discussions

Table I present the result of different group sizes of random grouping using SaNSDE as the optimizer. We can see that for fully separable functions f1-f3, random grouping can achieve competitive results when we choose the specified group size (e.g., group size 50 for f1, 10 for f2 and 30 for f3). Note that the result for f2 is much better than that of MOS [29] (the winner algorithm of CEC' 2013 competition on large scale optimization) with mean=8.32e+02 and std=4.48e+01, the result for f3 is as good as that of MOS, and the result for f1 is just slightly worse than that of MOS. These experiments give good reference to the LSGO algorithms with high accurate decomposition mechanism such as the formula based grouping [23, 24] and differential grouping [19–21], which decompose the problem by learning the variable interactions. When no interactions are detected, random grouping can help the algorithms achieve better performance in a few cases even for fully separable problems.

As for partially separable and non separable functions, the results of random grouping turns better as the group size get larger, the best results are achieved with no grouping except for f12 and f15. So we get the conclusion that when the variable interaction information is unknown (variable interaction detect technique can't be applied), the better choice is not to decompose the problem. Since the grouping size can affect the result, will random grouping achieve better result when using the same sizes of subgroups as in ideal grouping? To answer this question, two experiments are carried out.

Firstly, random grouping with the same group sizes of subgroups as those in ideal grouping is carried out, where "rand once" means the variables in the subgroups are fixed throughout the optimization while "rand each cycle" means the variables in the subgroups will be randomly regenerated for each cycle, the result is listed in table II. By analysis these results, we can see the following facts: 1) By comparing the last two columns of table II and the results in table I, it can be seen that random grouping which adopts the same sizes of subgroups as in ideal group can not get obviously better results; 2) it can be seen from the last two columns of table II that it is unnecessary for random grouping to change the subgroup each cycle, and 3) ideal grouping can help search algorithm to get better solutions than random grouping.

Secondly, comparison between ideal grouping and no grouping is made on f1-f11 (for f12-f15, ideal grouping is the same as no grouping since f12-f15 is fully non-separable) in table III. Table III shows a very startle result that the search algorithm with no grouping achieves better result than that with ideal grouping for fully separable functions f1-f3. However, by careful analysis, this unexpected result may be caused by the no efficiency enough of the search algorithm. In fact, if ideal grouping is used, the optimization of these fully separable 1000-D problems can be done by optimizing 1000 1-

D problems according to the definition of separability [30, 31]. As we all know, the optimization of 1000 one dimensional problems is usually much easer than optimization of one 1000-D problem. However, SaNSDE is now used as the line search algorithm to solve these one dimensional problems , but it is designed not for this purpose. Thus it is obviously inefficient as a line search algorithm. SaNSDE is designed for multiple variable problems and may generate better results with no grouping than it is used as a line search algorithm. This indicates that if the search algorithm is not efficient enough, then the decomposition method only has very limited effect on the optimization results.

To further verify this conclusion, the comparison between ideal grouping and the best random grouping is made on partially separable functions f4-f11, the results are listed in table IV. From table IV we can see that ideal grouping performs better than best random grouping. If we take a careful look at f4-f11 in table III, we can also see that ideal grouping performs a little better than no grouping. This reveals that after an effective variable grouping strategy is used, the key issue to get better solution is adopting efficient search algorithm for each subgroup. In summary, the results in these tables indicate that decomposition based on variable interaction does have good effect on generating better solution for LSGO, but it has to be integrated to an efficient search algorithm.

IV. Conclusion

In this paper, we investigate the effect of variable grouping and search algorithms on solution quality of large scale global optimization problems by comparing different grouping strategies. The experiment results on fully separable functions f1-f3 indicate even after the variables are grouped correctly, improperly choosing search algorithm will result in worse solutions. For fully separable problems, it is a key issue to use an efficient line search algorithm as the search algorithm. We also get the conclusion that if the search algorithm is not suitable or efficient enough, the decomposition based on variable interaction mechanism only plays limited role on solving the LSGO problems. For partially separable functions f4-f11, ideal grouping has better results than the best random grouping and no grouping. This indicates the correctly grouping variables is helpful to get better solutions. For fully non-separable functions f12-f15, ideal grouping (same as no grouping) achieves better performance than random grouping. This indicates that variable grouping strategy is useless for this kind of problems. Thus, we get the conclusion that decomposition mechanism based on variable interaction has a positive effect on the optimization. If suitable and efficient search algorithms are combined with the good decomposition mechanism, better results will be achieved. So, for LSGO problems, effective grouping strategy and efficient search algorithms are equally important. The best way to resolve LSGO problems can be in the following way: for fully separable functions, use efficient line search algorithm to optimize 1000 1-D problems; for partially separable problems, combine effective grouping strategy with efficient evolutionary algorithm; and for fully non-separable functions, do not use grouping strategy and use an efficient search algorithm to solve the whole problem.

The experiments also show that random grouping with same sizes of subgroups as those in the ideal grouping is not superior to it using random sizes of subgroups, and it is not necessary to change the subgroup randomly each cycle. How to design very efficient search algorithm and how to cooperate it with the good variable grouping strategy need further study in the future.

TABLE II
Comparisons of ideal grouping and random grouping on CEC' 2013 benchmark suit

P		ideal grouping	rand once	rand each cycle
f4	Best	**1.61e+08**	1.86e+13	7.91e+12
	Median	**2.77e+08**	3.86e+13	3.05e+13
	Worst	**5.92e+08**	5.31e+13	5.52e+13
	Mean	**2.93e+08**	3.71e+13	2.99e+13
	Std	**9.79e+07**	1.05e+13	1.39e+13
f5	Best	**1.58e+06**	6.20e+07	4.79e+07
	Median	**2.39e+06**	7.40e+07	6.23e+07
	Worst	**3.36e+06**	8.83e+07	7.79e+07
	Mean	**2.41e+06**	7.39e+07	6.17e+07
	Std	**5.72e+05**	7.20e+06	8.10e+06
f6	Best	**6.81e+04**	1.06e+06	1.04e+06
	Median	**1.27e+05**	1.06e+06	1.05e+06
	Worst	**1.76e+05**	1.07e+06	1.06e+06
	Mean	**1.27e+05**	1.06e+06	1.05e+06
	Std	3.17e+04	**3.31e+03**	4.01e+03
f7	Best	**5.24e+04**	2.28e+15	1.19e+14
	Median	**7.19e+04**	1.21e+16	6.89e+14
	Worst	**1.01e+06**	6.04e+16	5.31e+15
	Mean	**1.75e+05**	1.57e+16	1.15e+15
	Std	**2.50e+05**	1.52e+16	1.35e+15
f8	Best	**9.70e+12**	7.53e+17	6.14e+17
	Median	**5.54e+13**	2.02e+18	1.63e+18
	Worst	**1.40e+14**	3.67e+18	3.48e+18
	Mean	**5.82e+13**	2.06e+18	1.95e+18
	Std	**4.01e+13**	6.68e+17	8.91e+17
f9	Best	**1.31e+08**	4.93e+09	4.44e+09
	Median	**3.11e+08**	6.05e+09	5.44e+09
	Worst	**3.98e+08**	7.52e+09	7.22e+09
	Mean	**2.97e+08**	6.07e+09	5.61e+09
	Std	**7.16e+07**	7.19e+08	9.56e+08
f10	Best	**1.07e+02**	9.49e+07	9.21e+07
	Median	**1.34e+02**	9.57e+07	9.51e+07
	Worst	**2.33e+02**	9.64e+07	9.60e+07
	Mean	**1.45e+02**	9.58e+07	9.49e+07
	Std	**3.14e+01**	4.11e+05	9.55e+05
f11	Best	**1.29e+08**	1.70e+17	3.80e+16
	Median	**2.35e+08**	6.79e+17	3.56e+17
	Worst	**9.99e+10**	2.60e+18	1.58e+18
	Mean	**2.32e+10**	8.96e+17	5.46e+17
	Std	**3.64e+10**	6.78e+17	4.92e+17

Acknowledgment

This work was supported by National Natural Science Foundation of China (No.61472297 and No.U1404622) and the Fundamental Research Funds for the Central Universities (BDZ021430).

References

[1] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ, USA: IEEE Press, 1995.

TABLE I
RESULTS OF RANDOM GROUPING WITH DIFFERENT GROUP SIZES ON CEC' 2013 BENCHMARK SUIT

P		rand-10	rand-30	rand-50	rand-100	rand-150	rand-200	no-group
	Best	2.12e-14	6.63e-24	**3.18e-25**	8.79e-18	2.22e-12	2.12e-08	1.40e+04
	Median	5.75e-14	1.14e-23	**1.19e-24**	1.00e-16	1.46e-11	1.41e-06	1.20e+05
	Worst	1.32e-13	1.62e-23	**1.10e-23**	2.85e-16	6.57e-10	5.31e-05	2.09e+06
	Mean	5.96e-14	1.17e-23	**2.09e-24**	9.96e-17	8.09e-11	6.39e-06	3.20e+05
f1	Std	2.85e-14	**2.24e-24**	2.73e-24	7.48e-17	1.67e-10	1.36e-05	5.35e+05
	Best	2.42e-10	**5.57e-15**	3.22e-11	5.97e+00	1.25e+02	4.55e+02	1.34e+04
	Median	5.10e-10	**8.72e-15**	1.08e-10	2.59e+01	2.33e+02	6.67e+02	1.46e+04
	Worst	**2.31e-09**	9.95e-01	2.98e+00	1.68e+02	2.77e+02	1.38e+03	1.77e+04
	Mean	**7.43e-10**	6.63e-02	7.96e-01	3.86e+01	2.14e+02	7.71e+02	1.48e+04
f2	Std	**5.68e-10**	2.57e-01	1.01e+00	3.97e+01	4.86e+01	2.63e+02	1.22e+03
	Best	6.93e-07	4.76e-13	**1.99e-13**	1.93e+00	2.79e+00	3.34e+00	9.89e+00
	Median	1.01e-06	5.97e-13	**2.06e-13**	2.35e+00	3.75e+00	5.35e+00	1.08e+01
	Worst	1.83e-06	**7.67e-13**	9.92e-01	3.01e+00	4.36e+00	5.87e+00	1.28e+01
	Mean	1.09e-06	**5.96e-13**	1.26e-01	2.40e+00	3.78e+00	5.15e+00	1.09e+01
f3	Std	3.15e-07	**7.84e-14**	3.33e-01	3.06e-01	4.70e-01	6.44e-01	8.37e-01
	Best	5.59e+11	1.82e+11	9.28e+10	3.17e+10	2.19e+10	1.26e+10	**5.67e+09**
	Median	1.36e+12	3.70e+11	3.29e+11	7.78e+10	4.59e+10	2.36e+10	**6.92e+09**
	Worst	3.64e+12	1.24e+12	7.35e+11	1.44e+11	7.40e+10	7.01e+10	**1.12e+10**
	Mean	1.61e+12	4.83e+11	3.18e+11	7.55e+10	4.52e+10	3.23e+10	**7.61e+09**
f4	Std	7.94e+11	3.02e+11	2.12e+11	3.33e+10	1.72e+10	1.75e+10	**1.83e+09**
	Best	1.46e+07	1.38e+07	1.08e+07	5.56e+06	3.35e+06	2.59e+06	**2.58e+06**
	Median	2.17e+07	1.82e+07	1.40e+07	8.74e+06	6.21e+06	3.09e+06	**2.98e+06**
	Worst	2.82e+07	3.12e+07	1.67e+07	1.69e+07	7.77e+06	4.78e+06	**4.27e+06**
	Mean	2.16e+07	1.85e+07	1.36e+07	9.07e+06	5.99e+06	3.26e+06	**3.15e+06**
f5	Std	3.46e+06	4.33e+06	1.77e+06	2.96e+06	1.15e+06	6.24e+05	**5.20e+05**
	Best	9.78e+05	9.77e+05	3.69e+05	1.40e+05	1.16e+05	9.18e+04	**5.65e+03**
	Median	9.85e+05	9.84e+05	1.00e+06	1.93e+05	1.37e+05	1.09e+05	**9.12e+04**
	Worst	9.91e+05	9.92e+05	1.01e+06	2.62e+05	2.01e+05	1.42e+05	**1.39e+05**
	Mean	9.84e+05	9.84e+05	9.21e+05	1.92e+05	1.45e+05	1.09e+05	**8.99e+04**
f6	Std	3.96e+03	3.54e+03	2.17e+05	3.02e+04	2.58e+04	1.35e+04	**3.41e+04**
	Best	3.50e+09	8.45e+08	3.72e+08	4.12e+07	1.22e+07	1.29e+07	**4.03e+06**
	Median	8.02e+09	3.51e+09	1.28e+09	9.31e+07	4.44e+07	2.36e+07	**5.83e+06**
	Worst	3.27e+10	6.85e+09	2.40e+09	2.33e+08	9.22e+07	5.66e+07	**1.13e+07**
	Mean	1.11e+10	3.18e+09	1.35e+09	1.13e+08	4.68e+07	2.86e+07	**6.60e+06**
f7	Std	7.81e+09	1.83e+09	5.49e+08	6.45e+07	2.01e+07	1.33e+07	**2.03e+06**
	Best	2.67e+16	1.02e+16	3.11e+15	1.24e+15	4.57e+14	1.57e+14	**1.18e+13**
	Median	7.84e+16	2.87e+16	7.02e+15	2.82e+15	9.84e+14	6.45e+14	**2.13e+13**
	Worst	1.66e+17	6.93e+16	1.13e+16	9.74e+15	2.84e+15	1.21e+15	**4.75e+13**
	Mean	8.43e+16	3.20e+16	7.20e+15	3.33e+15	1.17e+15	6.44e+14	**2.22e+13**
f8	Std	4.04e+16	2.02e+16	2.02e+15	2.04e+15	6.47e+14	2.80e+14	**9.70e+12**
	Best	1.30e+09	8.84e+08	6.59e+08	4.37e+08	2.46e+08	**1.41e+08**	1.74e+08
	Median	1.57e+09	1.24e+09	1.15e+09	5.26e+08	4.22e+08	**2.52e+08**	2.71e+08
	Worst	2.32e+09	1.61e+09	1.44e+09	8.73e+08	6.83e+08	4.01e+08	**3.15e+08**
	Mean	1.63e+09	1.30e+09	1.04e+09	5.90e+08	4.23e+08	**2.63e+08**	2.70e+08
f9	Std	2.45e+08	2.37e+08	2.66e+08	1.41e+08	1.11e+08	7.14e+07	**3.42e+07**
	Best	8.20e+07	8.57e+07	5.08e+07	1.97e+07	1.11e+07	1.19e+07	**2.67e+03**
	Median	8.90e+07	8.86e+07	8.72e+07	3.00e+07	2.67e+07	1.61e+07	**4.83e+04**
	Worst	9.03e+07	9.01e+07	9.00e+07	8.91e+07	8.92e+07	2.30e+07	**4.89e+04**
	Mean	8.77e+07	8.83e+07	8.17e+07	4.50e+07	3.71e+07	1.63e+07	**3.95e+04**
f10	Std	2.79e+06	1.57e+06	1.16e+07	2.63e+07	2.86e+07	3.10e+06	**1.84e+04**
	Best	3.07e+11	1.02e+11	3.50e+10	9.18e+08	7.92e+08	2.37e+08	**2.94e+08**
	Median	8.73e+11	5.20e+11	1.70e+11	2.73e+10	2.53e+09	5.59e+08	**4.66e+08**
	Worst	1.99e+12	1.02e+12	5.11e+11	1.18e+11	4.62e+10	6.50e+09	**7.31e+08**
	Mean	1.00e+12	5.57e+11	2.27e+11	3.31e+10	1.69e+10	1.44e+09	**4.51e+08**
f11	Std	5.08e+11	2.95e+11	1.45e+11	3.30e+10	1.18e+10	2.06e+09	**1.20e+08**
	Best	5.55e+03	**1.48e+03**	2.57e+03	2.31e+03	2.42e+03	2.48e+03	3.45e+05
	Median	6.31e+03	**2.12e+03**	2.83e+03	2.79e+03	2.77e+03	2.82e+03	1.93e+06
	Worst	6.90e+03	**2.51e+03**	3.42e+03	3.39e+03	3.28e+03	3.11e+03	5.30e+06
	Mean	6.24e+03	**2.08e+03**	2.92e+03	2.79e+03	2.77e+03	2.83e+03	2.12e+06
f12	Std	3.92e+02	2.97e+02	2.48e+02	2.49e+02	2.22e+02	**1.67e+02**	1.50e+06
	Best	1.99e+10	1.19e+10	5.35e+09	2.29e+09	1.09e+09	6.20e+08	**2.17e+08**
	Median	3.89e+10	1.65e+10	1.05e+10	3.96e+09	2.37e+09	2.24e+09	**5.40e+08**
	Worst	8.96e+10	2.93e+10	2.02e+10	7.67e+09	5.12e+09	2.95e+09	**9.96e+08**
	Mean	4.29e+10	1.81e+10	1.04e+10	4.39e+09	2.66e+09	1.93e+09	**5.28e+08**
f13	Std	2.07e+10	5.47e+09	3.86e+09	1.68e+09	1.20e+09	6.87e+08	**2.24e+08**
	Best	3.88e+11	2.04e+11	1.22e+11	2.91e+10	1.56e+10	3.78e+08	**1.14e+08**
	Median	9.10e+11	3.50e+11	2.03e+11	6.13e+10	4.23e+10	1.40e+10	**4.24e+08**
	Worst	1.72e+12	5.24e+11	5.07e+11	1.62e+11	1.32e+11	3.44e+10	**1.32e+09**
	Mean	9.15e+11	3.87e+11	2.47e+11	7.23e+10	5.53e+10	1.50e+10	**5.35e+08**
f14	Std	3.52e+11	1.01e+11	1.03e+11	3.67e+10	3.65e+10	1.04e+10	**3.73e+08**
	Best	1.76e+08	8.80e+06	4.58e+06	1.95e+06	**1.57e+06**	1.71e+06	3.60e+06
	Median	3.23e+08	1.02e+07	4.95e+06	2.31e+06	3.43e+07	**2.04e+06**	4.93e+06
	Worst	1.17e+09	1.25e+07	**5.51e+06**	3.23e+07	3.92e+07	5.04e+07	1.30e+07
	Mean	3.88e+08	1.04e+07	**5.08e+06**	5.93e+06	2.28e+07	2.19e+07	5.66e+06
f15	Std	2.33e+08	1.04e+06	**2.92e+05**	9.78e+06	1.78e+07	2.22e+07	2.30e+06

TABLE III
COMPARISONS OF IDEAL GROUPING AND NO GROUPING ON CEC' 2013 BENCHMARK SUIT

P		ideal grouping	no grouping
f1	Best	1.09e+11	**1.40e+04**
	Median	1.54e+11	**1.20e+05**
	Worst	2.22e+11	**2.09e+06**
	Mean	1.52e+11	**3.20e+05**
	Std	2.64e+10	**5.35e+05**
f2	Best	3.61e+04	**1.34e+04**
	Median	4.64e+04	**1.46e+04**
	Worst	6.02e+04	**1.77e+04**
	Mean	4.76e+04	**1.48e+04**
	Std	6.56e+03	**1.22e+03**
f3	Best	1.98e+01	**9.89e+00**
	Median	2.00e+01	**1.08e+01**
	Worst	2.01e+01	**1.28e+01**
	Mean	2.00e+01	**1.09e+01**
	Std	6.36e-02	**8.37e-01**
f4	Best	**1.61e+08**	5.67e+09
	Median	**2.77e+08**	6.92e+09
	Worst	**5.92e+08**	1.12e+10
	Mean	**2.93e+08**	7.61e+09
	Std	**9.79e+07**	1.83e+09
f5	Best	**1.58e+06**	2.58e+06
	Median	**2.39e+06**	2.98e+06
	Worst	**3.36e+06**	4.27e+06
	Mean	**2.41e+06**	3.15e+06
	Std	5.72e+05	**5.20e+05**
f6	Best	6.81e+04	**5.65e+03**
	Median	1.27e+05	**9.12e+04**
	Worst	1.76e+05	**1.39e+05**
	Mean	1.27e+05	**8.99e+04**
	Std	**3.17e+04**	3.41e+04
f7	Best	**5.24e+04**	4.03e+06
	Median	**7.19e+04**	5.83e+06
	Worst	**1.01e+06**	1.13e+07
	Mean	**1.75e+05**	6.60e+06
	Std	**2.50e+05**	2.03e+06
f8	Best	**9.70e+12**	1.18e+13
	Median	5.54e+13	**2.13e+13**
	Worst	1.40e+14	**4.75e+13**
	Mean	5.82e+13	**2.22e+13**
	Std	4.01e+13	**9.70e+12**
f9	Best	**1.31e+08**	1.74e+08
	Median	3.11e+08	**2.71e+08**
	Worst	3.98e+08	**3.15e+08**
	Mean	2.97e+08	**2.70e+08**
	Std	7.16e+07	**3.42e+07**
f10	Best	**1.07e+02**	2.67e+03
	Median	**1.34e+02**	4.83e+04
	Worst	**2.33e+02**	4.89e+04
	Mean	**1.45e+02**	3.95e+04
	Std	**3.14e+01**	1.84e+04
f11	Best	**1.29e+08**	2.94e+08
	Median	**2.35e+08**	4.66e+08
	Worst	9.99e+10	**7.31e+08**
	Mean	2.32e+10	**4.51e+08**
	Std	3.64e+10	**1.20e+08**

TABLE IV
COMPARISON OF BEST RANDOM GROUPING AND IDEAL GROUPING ON CEC' 2013 BENCHMARK SUIT

P		rand-best	ideal-group
f4	Best	1.26e+10	**1.61e+08**
	Median	2.36e+10	**2.77e+08**
	Worst	7.01e+10	**5.92e+08**
	Mean	3.23e+10	**2.93e+08**
	Std	1.75e+10	**9.79e+07**
f5	Best	2.59e+06	**1.58e+06**
	Median	3.09e+06	**2.39e+06**
	Worst	4.78e+06	**3.36e+06**
	Mean	3.26e+06	**2.41e+06**
	Std	6.24e+05	**5.72e+05**
f6	Best	9.18e+04	**6.81e+04**
	Median	**1.09e+05**	1.27e+05
	Worst	**1.42e+05**	1.76e+05
	Mean	**1.09e+05**	1.27e+05
	Std	**1.35e+04**	3.17e+04
f7	Best	1.29e+07	**5.24e+04**
	Median	2.36e+07	**7.19e+04**
	Worst	5.66e+07	**1.01e+06**
	Mean	2.86e+07	**1.75e+05**
	Std	1.33e+07	**2.50e+05**
f8	Best	1.57e+14	**9.70e+12**
	Median	6.45e+14	**5.54e+13**
	Worst	1.21e+15	**1.40e+14**
	Mean	6.44e+14	**5.82e+13**
	Std	2.80e+14	**4.01e+13**
f9	Best	1.41e+08	**1.31e+08**
	Median	**2.52e+08**	3.11e+08
	Worst	4.01e+08	**3.98e+08**
	Mean	**2.63e+08**	2.97e+08
	Std	**7.14e+07**	7.16e+07
f10	Best	1.19e+07	**1.07e+02**
	Median	1.61e+07	**1.34e+02**
	Worst	2.30e+07	**2.33e+02**
	Mean	1.63e+07	**1.45e+02**
	Std	3.10e+06	**3.14e+01**
f11	Best	2.37e+08	**1.29e+08**
	Median	5.59e+08	**2.35e+08**
	Worst	**6.50e+09**	9.99e+10
	Mean	**1.44e+09**	2.32e+10
	Std	**2.06e+09**	3.64e+10

[2] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[3] 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, 2001, pp. 1101–1108 vol. 2.

[4] M. A. Potter and K. A. D. Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, pp. 1–29, 2000.

[5] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions:a survey of some theoretical and practical aspects of genetic algorithms," *Biosystems*, vol. 39, no. 3, pp. 263–278, 1995.

[6] K. Zhang, B. Li, and L. Tan, "Empirical study of the effect of variable correlation on grouping in cooperative coevolutionary evolutionary algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, June 2012, pp. 1–7.

[7] 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," 2010.

[8] B. Kazimipour, X. Li, and A. Qin, "Effects of population initialization on differential evolution for large scale

optimization," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 2404–2411.

[9] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the cec2013 special session and competition on large-scale global optimization," 2013.

[10] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, June 2008, pp. 1110–1116.

[11] ——, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985 – 2999, 2008, nature Inspired Problem-Solving.

[12] ——, "Multilevel cooperative coevolution for large scale optimization," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, June 2008, pp. 1663–1670.

[13] M. Omidvar, X. Li, Z. Yang, and X. Yao, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, July 2010, pp. 1–8.

[14] M. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, July 2010, pp. 1–8.

[15] J. Sun and H. Dong, "Cooperative co-evolution with correlation identification grouping for large scale function optimization," in *Information Science and Technology (ICIST), 2013 International Conference on*, March 2013, pp. 889–893.

[16] K. Weicker and N. Weicker, "On the improvement of coevolutionary optimizers by learning variable interdependencies," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3, 1999, pp. –1632 Vol. 3.

[17] 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*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6239, pp. 300–309.

[18] 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*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 1115–1122.

[19] M. 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.

[20] Y. Sun, M. Kirley, and S. K. Halgamuge, "Extended differential grouping for large scale global optimization with direct and indirect variable interactions," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015*, 2015, pp. 313–320.

[21] Y. Mei, M. N. OMIDVAR, X. Li, and X. Yao, "Competitive divide-and-conquer algorithm for unconstrained large scale black-box optimization," *ACM Transactions on Mathematical Software*, vol. V, no. N, 2015.

[22] N. Hansen, "The cma evolution strategy: A tutorial," 2005.

[23] F. Wei, Y. Wang, and T. Zong, "Variable grouping based differential evolution using an auxiliary function for large scale global optimization," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 1293–1298.

[24] ——, "A novel cooperative coevolution for large scale global optimization," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, Oct 2014, pp. 738–741.

[25] D. Molina, M. Lozano, and F. Herrera, "Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, July 2010, pp. 1–8.

[26] K. Tang, T. Weise, Z. Yang, X. Li, and P. N. Suganthan, "Results of the competition on high-dimensional global optimization at wcci2010," 2011.

[27] L.-Y. Tseng and C. Chen, "Multiple trajectory search for large scale global optimization," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, June 2008, pp. 3052–3059.

[28] S.-Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2175–2185, Nov. 2011.

[29] A. LaTorre, S. Muelas, and J.-M. Pena, "Large scale global optimization: Experimental results with mos-based hybrid algorithms," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 2742–2749.

[30] N. M. R. R. M. S. Anne Auger, Nikolaus Hansen, "Bio-inspired continuous optimization: the comming age," in *2007 IEEE congress on evolutionary computaion*, 2007.

[31] X. Li, "Decomposition and cooperative coevolution techniques for large scale global optimization," in *Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion*. New York, NY, USA: ACM, 2014, pp. 819–838.