Large-scale Cooperative Co-evolution with Bi-objective Selection Based Imbalanced Multi-Modal Optimization

Xingguang Peng
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, Shaanxi, P. R. China, 710072
pxg@nwpu.edu.cn

Yapei Wu
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, Shaanxi, P. R. China, 710072
wuyapei@mail.nwpu.edu.cn

***Abstract*—Cooperative co-evolutionary algorithm (CC) which runs in a divide-and-conquer manner is effective to solve large-scale global optimization (LSGO) problems. Multi-modal optimization (MMO) intends to locate multiple optimal solutions. Using MMO methods in CC algorithm would be beneficial, because MMO optimizer can provide more information about the landscapes. In this paper, a bi-objective selection is proposed to introduce imbalance among the subpopulations of a MMO optimizer. Only the highly representative subpopulations will be active and evolved in the MMO procedure. With this imbalanced MMO technique, the CC's subcomponents could obtain sufficient coevolutionary information (multiple optima) from each other. In addition, more computational resources could be saved and used in CC procedure. Experiments and statistical comparisons are conducted on LSGO benchmark functions to verify the effectiveness of the proposed method. The results indicate that the proposed algorithm significantly outperforms seven state-of-the-art large-scale CC algorithms.**

Keywords—Cooperative Co-evolutionary; large-scale optimization; Multi-Modal Optimization; bi-objective selection

I. INTRODUCTION

Optimization problems encountered in science and engineering have become increasingly complicated. On account of having a lot of decision variables and the complex interaction between variables, large-scale optimization (LSGO) problems are difficult to solve. It is necessary to develop effective methods for solving large-scale optimization problem [1].

A feasible approach to deal with large-scale optimization is the cooperative co-evolutionary (CC) algorithm. It decomposes the original complex problem into some subproblems which just acquire parts of the complete decision variables. Then each subproblem is optimized separately in a round-robin fashion. The above procedure is known as *divide-and-conquer* paradigm. By this means, the curse of dimensionality can be avoided. The validity of CC comes from the parallel computing and decomposing problem into simpler subproblems [2]. Compared with traditional evolutionary algorithms, CC algorithm brings two new tasks: the decomposition of the variables of original problem, and the cooperation between subproblems [3].

Since the large-scale problems are extremely complex and the landscape is intricate, most of meta-heuristic algorithms cannot avoid trapping in local optimum. Sometimes, it is difficult to even get a satisfactory solution. An intuitive method is to promote the diversity of population, thus larger space can be explored and the probability of obtaining the global optimum would be increased. As for CC algorithm, it is not enough to just increase the population diversity. One major reason is that the optimization of subproblems is essentially a dynamic optimization [4]. The evaluation of individuals in subproblem is related to collaboration information offered by other subproblems. In such uncertain environment, retain more information generated in optimization process will be beneficial [5]. Comparing with randomly increase the diversity of co-evolutionary populations, a promising approach would be to locate and maintain multiple optimal solutions simultaneously, since the superior solutions might have higher possibility to be the potential global optimal. Furthermore, multiple optimal solutions can assist reveal hidden properties of subproblems [6], which is positive for the CC to converge to the global best Nash equilibrium (global optimum) [4].

The multi-modal optimization (MMO) is a technique that simultaneously search and maintain multiple optima in the landscapes. In the area of evolutionary computation (EC), niching based approaches are popularly employed to obtain multiple optima. However, to introduce the MMO technique into the CC framework, computation budget should be carefully considered. This is because in the niching based MMO, several subpopulations are generated by niching techniques, each of them have to be well evolved to find corresponding optima. Especially in a relative large landscapes (or relative high-dimensional search space), a large overall population is required to conduct sufficient multi-modal optimization. Obviously, this is an intensive computational resource consuming procedure. Considering the limited computational resources, a MMO based CC might not conduct the CC process sufficiently to find the global optimum.

From the perspective of CC, the MMO optimizer do not have to be fully conducted for each subcomponent of CC to well evolve all of the potential optima. Just the ones with good fitness values and diversity are sufficient to help the subcomponents co-evolve with each other. In this work, an

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

imbalanced MMO with bi-objective (fitness and diversity) selection is proposed and introduced into the CC framework. The bi-objective selection is used to introduce imbalance among the subpopulations of a MMO optimizer. Only the highly representative subpopulations will be active and evolved in the MMO procedure. With this imbalanced MMO technique, the CC's subcomponents could obtain sufficient coevolutionary information (multiple optima) from each other. In addition, more computational resources could be saved and used in CC procedure. A series of experiments are carried out on the CEC'10 LSGO competition benchmark suite to verify the effectiveness of the proposed method.

The rest of the paper is organized as follows. In section II, we have a brief reviews on the CC framework. The combination of imbalanced MMO method and CC framework is proposed in Section III. In Section IV, some experimental results are shown, and the comparisons with other CC algorithms are analyzed. Conclusion and future work are drawn in Section V.

II. COOPERTIVE CO-EVOLUTION

Cooperative co-evolution conducts a *divide-and-conquer* strategy. Before proceeding the optimization, it decomposes the original problem into many subproblems, which are low dimensional and easier to optimize. The *divide-and-conquer* framework enable us to solve large scale problems efficiently.

In order to implement the framework, many decomposition methods, like delta grouping [7] and differential grouping [8], are proposed sequentially. The grouping methods are mainly classified into two categories: fixed grouping and dynamic grouping. For example, one-dimensional method [9] divides a D-dimension problem into D subproblems, hence, every subproblem is one-dimension. Another typical fixed grouping method is to divide a D-dimension problem into k S-dimension subproblems [10]. The parameter k and S is predefined. Although these fixed grouping methods are not very adaptive, it is viable and simple to implement.

Differential grouping is a state-of-art grouping method which can divide the variables of original problem into different subproblems automatically. It is a dynamic grouping method. By detecting the interaction between variables, dependent variables are grouped into the same subproblem, thus keeping the correlation between subproblems at low levels. Research [11] shows that interdependence between variables have influence on the performance of optimization algorithms.

Algorithm 1: CC algorithm

1: Group the original problem into *N* subproblems
2: Randomly initialize the context vector *CV*
3: Initialize *SP(i)*, *i=1,….N*
4: **while** *Termination=false*
5:    **for** *i=1:N*
6:       *(CV)=optimizer(SP(i),CV)*
7:    **end**
8: **end**
9: *Best=CV*
10: **Return** *Best*

The pseudo-code of general CC framework is shown in Algorithm 1. At first, the original problem is decomposed into many subproblems with one grouping method, then the context vector is initialized. Every subproblem merely is one part of the original problem. Therefore, we need a context vector (CV) to evaluate individuals in subproblems. Usually, the CV is initialized as the best solution in initial population. After initializing the subproblem populations (SP), subproblems will be optimized in a round-robin fashion. The optimizer could be any evolutionary algorithm. The historical best solution of each subproblem is often stored as the CV. The algorithm terminates when run out of the fitness evaluation budget. In line 10, the best solution is output as final result.

III. PROPOSED METHODS

In this section, the idea and implementation of CC algorithm with imbalanced multi-modal optimizer (IMMO-CC) is described in detail. The framework of IMMO-CC, and two newly absorbed procedures, bi-objective selection and multi-modal optimizer will be introduced.

A. Algorithm Framework

The general description of proposed framework is given as follows. First, the original problem is decomposed into N subproblems (or the decision variables are grouped into N groups) using a given grouping method. Then, the imbalanced multi-modal optimizer is employed to optimize each subproblem. Considering the amount of computational resources consumed by the multi-modal optimizer is relative large, a bi-objective selection is carried out in the multi-modal optimizer to select highly representative subpopulations to be active during the MMO procedure. The inactive subpopulations will freeze for one iteration, but they still could be selected out in next iteration.

The schematic diagram of IMMO-CC is exhibited in Fig. 1. As for each subproblem, M subpopulations (M is not fixed for each subproblem and is determined by the corresponding MMO optimizer) are maintained in separate regions of the search space to seek for multiple optimal solutions. These solutions are exchanged among the subproblems to conduct co-evolution. The FEs in Fig. 1 means the number of fitness evaluations.

B. Multimodal Optimizer

The goal of MMO methods is to find multiple optimal solutions simultaneously [6]. MMO method is also called niching method. Niching methods usually maintain some subpopulations in different areas of the search space of a subproblem. Many niching methods have been proposed, including fitness sharing [13], crowding methods [14], derating [15], restricted tournament selection [16,17], and speciation [18,19].

In this study, a covariance matrix self-adaptation with repelling subpopulation (RS-CMSA) [20] is introduced as the multi-modal optimizer to work in our proposed IMMO-CC algorithm. RS-CMSA has demonstrate unparalleled performance on solving multi-modal optimization problems.

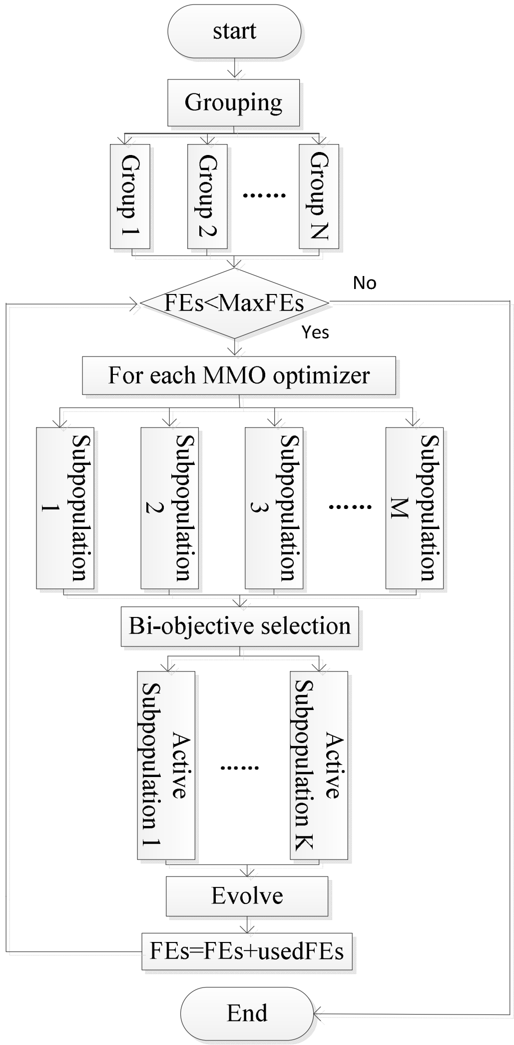

Fig. 1. IMMO-CC schematic diagram

RS-CMSA consists of a covariance matrix self-adaptation evolution strategy(CMA-ES) [21] and a niching strategy with taboo points. The CMA-ES is a powerful optimizer. It converges to optimal solution, by sampling and updating its parameters (step size, covariance, center) with sample information. CMA-ES shows an excellent performance on continuous optimization problems.

The main niching idea of RS-CMSA is that define and archive taboo points and taboo distance to prevent population converging in the same area. To seek for multiple optimal solutions, the population is divided into several subpopulations. These subpopulations evolve in parallel and each subpopulation has their own parameters. The high-level pseudo-code of RS-CMSA is shown in Algorithm 2.

C. *Imbalance: a bi-objective selection*

The subpopulations which not only have higher fitness but also diversity should be distribute more computational resources. To select out such subpopulation, a bi-objective optimization problem is formulated. The first objective is the fitness of subpopulations, another one is the diversity of subpopulations.

Algorithm 2 High-level pseudo-code of RS-CMSA

1: Initiate N subpopulation center in search space;
2: Initiate taboo points and taboo distances;
3: **while** Not terminated;
4: Produce M taboo acceptable individuals in each subpopulation;
5: Adjust parameters of subpopulations according to new individuals;
6: Update taboo points and taboo distance;
7: **end**
8: Output;

In this paper, the diversity of one subpopulation is defined as its average Manhattan distance with the other subpopulations. Manhattan distance is simple to compute, and can indicate similarity between vectors. The Manhattan distance of date point x_i and x_j is:

$$d_{ij} = \sum_{k=1}^{n} \left| x_i(k) - x_j(k) \right|$$

where n is the dimension of data, $x_i(k)$ is the k-th dimension of data x_i. The diversity of x_i is measured as:

$$D(x_i) = \frac{1}{N_p} \sum_{k=1, k\neq j}^{N_p} d_{ik}$$

where N_p is the number of subpopulation.

The bi-objective selection is defined as to select subpopulation:

$$\arg\min\left\{-D(x_i), f(x_i)\right\}$$

The nondominated sorting method used in NSGA-II [12] is employed to distribute subpopulations into different nondominated fronts. The subpopulations in the first front are selected out to evolve in the next iteration.

D. *Constructing Complete Solutions for Fitness Evaluation*

The main difference between classical evolutionary algorithms (EAs) and CC algorithms is that in CC algorithms an individual encodes only a segment (according to its subproblems) of the solution. To evaluate such a solution segment, one has to collect collaborators from the other subpopulations to construct complete solutions for fitness calculation. In the conventional CC algorithms, only the best solutions of each subproblem are exchanged and mixed together to construct the complete solution for fitness evaluation. However, in this paper, each subproblem may more than one optima to the other subproblems. Therefore, the strategy of constructing complete solutions should be specifically designed to avoid the combinatorial explosion.

In this paper, we also use the bi-objective selection proposed above to select a reasonable number of collaborators to construct complete solutions. More particularly, the subcomponents provide their collaborators (multiple optima) together with the complete solution contexts and fitness values. In addition, the historical best solution of a subproblem is also maintained in terms of complete solution. As for a certain subproblem, when constructing complete solutions, the received collaborators are considered together with its historical best solution. Among these solutions, the highly representative collaborators are selected according to their fitness values and diversity. With the selected collaborators, an individual in the subproblem can be calculated by replacing the corresponding solution segments of the collaborators with this individual. Among the resulting complete solutions, the fittest one is used to evaluate the individual.

IV. EXPERIMENTS

In this section, experiments are conduct on the CEC'10 LGSO benchmark function. Comparison results with other popular LSGO algorithm are analyzed.

A. Experimental setting

Each algorithm has performed 25 independent runs. In each run, algorithm stops when the number of FEs exceeds 3×10^6.

The general settings of IMMO-CC are as fellows. We choose the excellent differential grouping (DG) as the grouping method of IMMO-CC. The value of the parameter in DG is assigned as 10^{-6}. The number of subpopulations of IMMO optimizer is set to 10. The evolutional generations of each subproblem is set to 100.

B. Performance Comparison

Some experiments are conducted on the 20 1000-dimentional benchmark functions launched on CEC'10 LSGO competition. Those benchmark functions can be classified to four categories:

1. *fully separable (F1～F3);*
2. *partially separable with a single 50-dimensional nonseparable group (F4～F8);*
3. *partially separable with 10 50-dimensional nonseparable groups (F9～F13);*
4. *partially separable with 20 50-dimensional nonseparable groups(F14～F18);*
5. *fully nonseparable(F19～F20) functions;*

The detail information of the benchmark functions can be seen in [22]. Seven state-of-the-art large scale CC algorithms (briefly descripted in Table 1) are used to compare with the proposed IMMO-CC.

Table 2 shows the statistical results of mean, standard deviation and *p-value*. A left-tailed Wilcoxon signed ranks test is conducted with 0.05 significant level. The null hypothesis is that the IMMO-CC has significant better performance than the corresponding algorithm. The null hypothesis is true when the *p-value* is smaller than 0.05. It can be seen in Table 2 that the IMMO-CC performs better on most functions compared with other 7 algorithms.

TABLE 1. Description of the compared large-scale CC algorithms.

Algorithm	Grouping method	Optimizer
DECC-D [7]	Delta grouping	SaNSDE [25]
MLCC [23]	Random grouping with a pool of potential subcomponent sizes	
DECC-DML [7]	Delta grouping with a pool of potential subcomponent sizes	
DECC-DG [8]	Differential grouping (DG)	
DECC-I [8]	Ideal grouping	
CBCC-1	DECC-DG with contribution based CC algorithm [24]	CMA-ES
CBCC-2		

The last row in Table 2 is the number of significantly better results. The first number is the number of better results obtained by the IMMO-CC, another number is the number of better results obtained by other algorithm. The proposed IMMO-CC get 9 overall better results compared with all the other algorithm. Furthermore, Table 3 show detail information about the comparisons between IMMO-CC and other algorithms. These results demonstrate that the proposed IMMO-CC have better results in 99 out of 140 comparisons, and worse results in 36 out of 140 comparisons. The experiment results verify the proposed IMMO-CC is an effective algorithm.

V. CONCLUSIONS

In this study, a CC algorithm with imbalance multi-modal optimizer is proposed. Since CC decomposes large scale problem into many subproblems, imbalanced multi-modal optimizer can help reveal relations between subproblems by locating multiple optimal simultaneously. In order to efficiently use the computation resources, a bi-objective selection is carried out in multi-modal optimizer. By the bi-objective selection, subpopulations which have better fitness and high diversity are selected out. As a consequence, far more better results are achieved by the proposed method.

Experiments on CEC'10 LSGO benchmark have been carried out. The experiment results and statistical analysis show that the proposed method significantly outperforms other 7 outstanding CC algorithm. The results have verified the validity of the idea that utilizing multi-modal optimizer in CC framework to solve LSGO problems.

When using multi-modal optimizer in CC framework, every few studies have paid attention to taking advantage of the multiple optimal maintained by multi-modal optimizer. Future work can concentrate on improving the collaboration between subcomponents of CC algorithm by using landscape information obtained by multi-modal optimizer.

ACKNOWLEDGMENT

This work has been supported by National Science Foundation of China (No. 61473233), Fundamental Research Funds for the Central Universities of China under Grant 3102016ZY007.

TABLE 2. Performance comparisons on CEC’10 LSGO benchmark function

Functions		DECC-D	MLCC	DECC-DML	DECC-DG	DECC-I	CBCC-1	CBCC-2	IMMO-CC
F1	Mean	9.60E-21	8.24E-07	2.77E-07	2.08E+06	3.83E+05	1.96E+06	6.38E+06	6.23E-02
	Std	9.11E-21	4.28E-07	9.60E-07	2.05E+06	6.51E+05	1.99E+06	1.82E+07	6.93E-03
	p-value	1.00E+00	1.00E+00	1.00E+00	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F2	Mean	6.52E+01	2.61E-03	1.04E+01	4.22E+03	4.39E+03	4.33E+03	4.18E+03	1.25E+03
	Std	4.47E+01	5.34E-03	2.24E+01	3.80E+02	2.96E+02	3.04E+02	5.38E+02	4.74E+01
	p-value	1.00E+00	1.00E+00	1.00E+00	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F3	Mean	2.29E+00	1.27E-02	2.57E-01	1.09E+01	1.10E+01	1.12E+01	1.10E+01	1.36E+00
	Std	1.75E-01	2.65E-02	7.06E-01	8.53E-01	6.23E-01	8.96E-01	7.32E-01	1.09E-01
	p-value	**6.54E-06**	1.00E+00	1.00E+00	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F4	Mean	2.98E+12	1.17E+14	1.18E+14	5.06E+11	2.71E+10	1.81E+11	1.65E+10	9.83E+08
	Std	9.35E+11	4.12E+13	1.69E+14	1.96E+11	1.24E+10	1.08E+11	3.62E+09	4.17E+08
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F5	Mean	2.86E+08	5.04E+08	4.99E+08	7.36E+07	6.86E+07	7.02E+07	6.43E+07	8.20E+07
	Std	1.08E+08	1.36E+08	1.28E+08	9.56E+06	1.24E+07	1.05E+07	1.31E+07	1.42E+07
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	9.75E-01	9.95E-01	9.97E-01	9.99E-01	/
F6	Mean	5.89E+06	1.90E+07	1.68E+07	1.58E+01	1.63E+01	8.14E+04	4.11E+04	9.08E-01
	Std	5.43E+06	2.12E+06	6.08E+06	7.30E-01	9.69E-01	2.84E+05	2.05E+05	1.80E-01
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F7	Mean	1.47E+05	4.88E+10	3.42E+10	2.79E+04	1.17E+04	1.23E+05	1.26E+10	6.60E+09
	Std	2.47E+05	1.64E+10	5.19E+10	2.03E+04	3.96E+03	1.09E+05	1.48E+10	7.90E+09
	p-value	1.00E+00	**6.54E-06**	**4.25E-02**	1.00E+00	1.00E+00	1.00E+00	**4.25E-02**	/
F8	Mean	1.27E+08	8.23E+08	3.10E+10	2.78E+07	8.06E+05	7.50E+06	3.72E+07	9.72E+07
	Std	1.52E+08	1.92E+08	6.90E+10	3.19E+07	1.63E+06	1.84E+07	3.47E+07	3.49E+07
	p-value	4.79E-01	**6.54E-06**	**9.37E-02**	1.00E+00	1.00E+00	1.00E+00	1.00E+00	/
F9	Mean	1.01E+08	1.69E+09	1.05E+09	3.65E+07	4.76E+07	1.02E+07	3.40E+08	4.98E+07
	Std	9.09E+06	2.54E+08	1.13E+09	1.49E+07	5.30E+07	3.84E+06	2.67E+08	3.58E+07
	p-value	**9.42E-06**	**6.54E-06**	**6.54E-06**	8.97E-01	7.49E-01	1.00E+00	**6.54E-06**	/
F10	Mean	4.07E+03	5.19E+03	4.30E+03	3.33E+03	3.13E+03	2.59E+03	4.90E+03	1.57E+03
	Std	1.26E+03	1.72E+03	1.77E+03	1.92E+02	1.68E+02	1.48E+02	6.37E+02	9.15E+01
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F11	Mean	9.98E+01	2.00E+02	1.91E+02	2.64E+01	2.51E+01	2.69E+01	2.75E+01	1.81E+00
	Std	1.01E+02	2.24E+00	3.56E+01	2.95E+00	2.72E+00	2.64E+00	3.18E+00	7.65E-01
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F12	Mean	9.14E+03	8.68E+05	4.76E+05	3.21E+04	2.44E+04	3.53E+04	5.07E+04	4.16E-01
	Std	1.08E+03	1.24E+05	4.69E+05	1.06E+04	7.12E+03	1.11E+04	1.10E+04	2.66E-01
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F13	Mean	5.44E+03	3.24E+04	8.62E+04	2.89E+07	1.29E+04	9.06E+04	1.29E+07	1.05E+04
	Std	2.76E+03	2.61E+04	1.95E+05	1.57E+07	4.34E+03	6.11E+04	7.36E+06	5.67E+03
	p-value	9.99E-01	**1.35E-05**	9.19E-01	**6.54E-06**	**4.25E-02**	**6.54E-06**	**6.54E-06**	/
F14	Mean	3.00E+08	3.62E+09	2.22E+09	2.10E+07	2.14E+07	2.24E+07	5.35E+09	1.82E+08
	Std	2.19E+07	5.43E+08	2.04E+09	2.25E+06	2.06E+06	2.27E+06	6.00E+08	3.63E+07
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	1.00E+00	1.00E+00	1.00E+00	**6.54E-06**	/
F15	Mean	1.30E+04	1.17E+04	1.10E+04	2.88E+03	2.84E+03	2.84E+03	3.22E+03	1.74E+03
	Std	2.18E+02	2.05E+03	2.77E+03	2.76E+02	1.86E+02	2.65E+02	4.17E+02	1.62E+02
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F16	Mean	2.02E+02	3.99E+02	3.62E+02	1.97E+01	1.93E+01	1.87E+01	1.91E+01	1.47E+00
	Std	1.58E+02	3.43E+00	1.09E+02	3.61E+00	3.77E+00	3.83E+00	2.76E+00	1.53E+00
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F17	Mean	7.47E+04	1.79E+06	9.71E+05	7.76E+00	7.08E+00	1.49E+01	1.24E+02	1.55E-01
	Std	4.72E+03	1.78E+05	1.05E+06	1.89E+00	1.76E+00	7.01E+00	5.72E+01	1.72E-02
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F18	Mean	1.44E+04	1.07E+05	7.77E+04	2.01E+10	1.15E+03	4.10E+09	1.23E+11	1.96E+06
	Std	1.27E+04	2.68E+04	1.75E+05	4.82E+09	1.65E+02	1.83E+09	1.45E+10	2.06E+06
	p-value	1.00E+00	1.00E+00	1.00E+00	**6.54E-06**	1.00E+00	**6.54E-06**	**6.54E-06**	/
F19	Mean	1.59E+06	2.96E+06	2.70E+06	9.01E+05	8.95E+05	9.12E+05	9.11E+05	6.06E+03
	Std	1.32E+06	4.29E+05	3.37E+06	6.14E+04	6.24E+04	7.11E+04	6.02E+04	8.60E+02
	p-value	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	**6.54E-06**	/
F20	Mean	2.27E+03	1.75E+05	5.42E+03	6.53E+08	1.67E+07	1.41E+07	6.97E+09	8.91E+10
	Std	2.44E+02	2.08E+05	1.46E+04	6.71E+08	3.30E+07	1.96E+07	1.12E+09	7.45E+10
	p-value	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	/
Significant better No.		13 vs 5	15 vs 5	14 vs 5	14 vs 4	13 vs 5	14 vs 5	17 vs 2	Overall Best No.9

TABLE 3. Summary of statistical significant comparisons. Significant level is 0.05. '+': the proposed method is significant better, '-':the proposed method is significantly wore, '～': not significantly different.

	Function number																			
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
DECC-D	-	-	+	+	+	+	-	～	+	+	+	+	-	+	+	+	+	-	+	-
MLCC	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	-
DECC-DML	-	-	-	+	+	+	+	～	+	+	+	+	～	+	+	+	+	-	+	-
DECC-DG	+	+	+	+	-	+	-	-	～	+	+	+	+	-	+	+	+	+	+	-
DECC-I	+	+	+	+	-	+	-	-	～	+	+	+	+	-	+	+	+	-	+	-
CBCC-1	+	+	+	+	-	+	-	-	-	+	+	+	+	-	+	+	+	+	+	-
CBCC-2	+	+	+	+	-	+	+	-	+	+	+	+	+	+	+	+	+	+	+	-

REFERENCES

[1] Kazimipour, B, et al. "A novel hybridization of opposition-based learning and cooperative co-evolutionary for large-scale optimization." *Evolutionary Computation* IEEE, 2014:2833-2840.

[2] Liu, Y., Yao, X., Zhao, Q., & Higuchi, T. (2001). Scaling up fast evolutionary programming with cooperative coevolution. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (Vol.2, pp.1101-1108 vol. 2). IEEE.

[3] Mei, Yi, et al. "A Competitive Divide-and-Conquer Algorithm for Unconstrained Large-Scale Black-Box Optimization." *Acm Transactions on Mathematical Software* 42.2(2016):13..

[4] Peng, Xingguang, K. Liu, and Y. Jin. "A dynamic optimization approach to the design of cooperative co-evolutionary algorithms." Knowledge-Based Systems 109(2016):174-186.

[5] Peng, Xingguang, X. Lei, and K. Liu. "Compensate information from multimodal dynamic landscapes: An anti-pathology cooperative coevolutionary algorithm." Evolutionary Computation IEEE, 2014:2578-2584.

[6] Li, Xiaodong, et al. "Seeking multiple solutions: an updated survey on niching methods and their applications." *IEEE Transactions on Evolutionary Computation* (2016).

[7] Omidvar, Mohammad Nabi, Xiaodong Li, and Xin Yao. "Cooperative co-evolution with delta grouping for large scale non-separable function optimization." *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010.

[8] Omidvar, Mohammad Nabi, et al. "Cooperative co-evolution with differential grouping for large scale optimization." *IEEE Transactions on Evolutionary Computation* 18.3 (2014): 378-393.

[9] Potter, Mitchell A., and Kenneth A. De Jong. "A cooperative coevolutionary approach to function optimization." *International Conference on Parallel Problem Solving from Nature*. Springer Berlin Heidelberg, 1994.

[10] Van den Bergh, Frans, and Andries Petrus Engelbrecht. "A cooperative approach to particle swarm optimization." *IEEE transactions on evolutionary computation* 8.3 (2004): 225-239.

[11] Salomon, Ralf. "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms." *BioSystems* 39.3 (1996): 263-278.

[12] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2 (2002): 182-197.

[13] Goldberg, David E, and J. Richardson. "Genetic algorithms with sharing for multimodal function optimization." *International Conference on Genetic Algorithms, Cambridge, Ma, Usa, July* DBLP, 1987:41-49.

[14] Kenneth A.. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Diss. 1975.

[15] Beasley, David, David R. Bull, and Ralph R. Martin. "A sequential niche technique for multimodal function optimization." *Evolutionary computation* 1.2 (1993): 101-125.

[16] Harik, Georges R. "Finding Multimodal Solutions Using Restricted Tournament Selection." *ICGA*. 1995.

[17] Roy, R., and Ian C. Parmee. "Adaptive restricted tournament selection for the identification of multiple sub-optima in a multi-modal function." *AISB Workshop on Evolutionary Computing*. Springer Berlin Heidelberg, 1996.

[18] Pétrowski, Alain. "A clearing procedure as a niching method for genetic algorithms." *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 1996.

[19] Li, Jian-Ping, et al. "A species conserving genetic algorithm for multimodal function optimization." *Evolutionary computation* 10.3 (2002): 207-234.

[20] Ahrari, Ali, Kalyanmoy Deb, and Mike Preuss. "Multimodal Optimization by Covariance Matrix Self-Adaptation Evolution Strategy with Repelling Subpopulations." *Evolutionary computation* (2016).

[21] Hansen, N, and A. Ostermeier. "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation."*IEEE International Conference on Evolutionary Computation* IEEE Xplore, 1996:312-317.

[22] Tang, Ke, et al. "Benchmark Functions for the CEC 2010 Special Session and Competition on Large-Scale Global Optimization (2010)."*Nature Inspired Computation and Applications Laboratory, USTC, China*.

[23] Z. Yang, K. Tang, X. Yao, "Multilevel cooperative coevolution for large scale optimization," *IEEE Congress on Evolutionary Computation (CEC)*, 2008, pp. 1663‑1670.

[24] Omidvar, Mohammad Nabi, Xiaodong Li, and Xin Yao. "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms." *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 1115-1122.

[25] Yang, Zhenyu, Ke Tang, and Xin Yao. "Self-adaptive differential evolution with neighborhood search." *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 2008, pp. 1110-1116.