Cooperative Co-evolution for Large Scale Optimization with Dynamic Variable Grouping via Marginal Product Modeling

Yapei Wu
School of Marine Science and Technology
Northwestern Polytechnical University
Xian, P. R. China
wuyapei@mail.nwpu.edu.cn

Xingguang Peng*
School of Marine Science and Technology
Northwestern Polytechnical University
Xian, P. R. China
pxg@nwpu.edu.cn

Demin Xu
School of Marine Science and Technology
Northwestern Polytechnical University
Xian, P. R. China
xudm@nwpu.edu.cn

***Abstract*—Cooperative co-evolution (CC) algorithm is a promising method to deal with large scale optimization (LSGO) problem. One major challenge in CC is to design a good decomposition strategy to decompose original problem into subproblems. Linkage learning is a technique to search the linkage betweens decision variables. In this paper, we proposed a dynamic online grouping CC algorithm which employ the linkage learning method of extended compact genetic algorithm (ECGA). This linkage learning method build marginal product models of population and search for models according to minimum description length (MDL) criterion. A discretization technique and normalization method of MDL value is used to make the linkage learning more adaptive in large scale optimization. The decision variables will be regrouped according to the linkage learned when the optimization procedure is considered as stagnant. Experiments are conduct on CEC'10 LSGO benchmark function. The test results confirm the validity of this method.**

***Index Terms*—Large scale optimization, cooperative co-evolution, linkage learning, variable grouping**

I. Introduction

With the development of technology, good methods to solve large-scale global optimization (LSGO) problem is needed, especially in artificial intelligence industry. For example, there are hundreds of millions of parameters in deep neural networks, optimizing these parameters to find a good network settings has significant impact on network performance [1] [2]. With the rise of artificial intelligence, the research on large-scale global optimization has received growing attention in recent years.

Cooperative Co-evolution (CC) is considered as an effective method to solve large-scale optimization. The framework of the co-evolutionary algorithm was first proposed by Potter and De Jong [3]. According to the strategy of *divide and conquer*, the problem is first decomposed into subproblems and then subproblems are optimized respectively . The *divide and conquer* strategy brings CC the ability to solve large-scale optimization problems, but also brings two new challenge: how to design the decomposition strategy (also called grouping), and how to design a collaborative strategy between subproblems.

*Corresponding author

Many decomposition strategies have been proposed in recent years. These method can be classified into two categories: offline grouping and dynamic online grouping. The risk of offline grouping is that once the interactive variables are mistakenly divided into different subproblems, there will be no chance to correct. Differential grouping (DG) [4] is a typical offline method. It decomposes the decision variables into groups before optimization procedure, and the group structure will stay unchanged until the algorithm finishes. As for dynamic online decomposition method, the group structure changes in optimization procedure, which can be more adaptive for different problems, especially for nonseparable problems and dynamic problems. The typical dynamic online method are random grouping [5] and delta grouping [6].

The decomposition strategy in CC is similar to linkage learning in genetic algorithm. The goal of linkage learning is to identify the blocks of genes that need to stay together in crossover operation. Extended compact genetic algorithm (ECGA) [7] is a kind of genetic algorithm which can learn the linkage of gene. It use the product of marginal distribution to build model of population. ECGA learn the linkage between genes by searching a simple marginal product models (MPM) which can explain the current population well.

Inspired by the linkage learning method in ECGA, we propose a CC algorithm which use the linkage learning method in ECGA as variables grouping strategy. A discretization technique of decision variables and normalization method of the evaluation criterion of the models are proposed to make the linkage learning method effective for large-scale continuous optimization. Experiments are carried out on CEC'10 LSGO benchmark functions to verify the effectiveness of the proposed method.

The rest of the paper is organized as follows. In Section II, we briefly introduce CC and ECGA. The proposed method is described in detail in Section III. In Section IV, experimental results are demonstrated and the comparison with other CC algorithm is conducted. Section V concludes the paper.

II. Preliminaries

A. Cooperative Co-evolution

CC algorithm runs in a *divide and conquer* manner, which is a promising approach to deal with large-scale optimization problem. Fig.1 shows the diagram of CC algorithm. As can be seen from Fig.1, CC algorithm mainly includes three processes: decomposition, optimization and collaboration. Besides optimization, choosing a good decomposition strategy and designing a proper collaborative strategy are all crucial to the performance of CC algorithm.

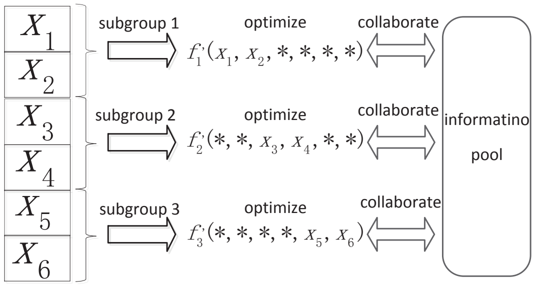

Fig. 1. schematic of CC framework

Literature [8] pointed out that the interdependence between variables will greatly affect the performance of the optimization algorithm. Ideally, grouping strategy aims to minimize the interaction between subproblems. Many decomposition method have been proposed, such as random grouping [5], delta grouping [6] and differential grouping [4]. DG decomposes problem by detecting the pairwise interactions between decision variables, and it results in a high grouping precision. Recently, some improved method based on DG were proposed. Mei *et al.* [9] proposed global differential grouping (GDG), which use global information of the interactions between variables. A recursive differential grouping (RDG) method is proposed by Sun *et al.* [10], and DG2 which is a improved variant of DG is proposed by Omidvar *et al.* [11]. These three methods improved DG in grouping accuracy and efficiency. Developing new grouping strategy and increasing the accuracy of variable grouping is still one important aspect of the research on CC.

The literature [12] found that after the problem was decomposed, many of the important evolutionary information were lost because the subproblems had only part variables of the original problem. Without such information, the evaluation of subproblem individual fitness tends to be ”one-sidedness”, which can not guarantee that the problem can converge to the global optimal solution. This phenomenon exists in inaccurately decomposed problems and nonseparable problems. Therefore, it is also an interesting topic of the CC algorithm to improve the collaborative strategy between the subproblems to increase the probability of reaching global optimum.

In the traditional collaborative strategy of subproblems, each subproblem share its current optimal solution as cooperation information. Such collaborative strategy may lead algorithm to trap into local optimum. Literature [13] [14] improves the subproblems' collaborative strategy of CC by searching for multiple optimal solutions simultaneously in subproblems and putting multiple optimal solutions into cooperative information pool. In this way, it ensures the quality of cooperative information and increasing the diversity of cooperative information. Such information compensation schema will help the co-evolution algorithm jump out of the local optimum and increase the probability of finding the global optimal solution.

B. Extended Compressed Genetic Algorithm

Extended Compressed Genetic Algorithm (ECGA) is based on the idea that the probability distribution can be used to model a population of genetic algorithms. Obtaining a good probability distribution model for population description is equivalent to learning linkage between variables. The probabilistic models used in ECGA are marginal product models (MPMs) [15], which is the product of the marginal probability distributions of different variables. The models are evaluated by the Minimum Description Length (MDL) [16] criterion. The essence of MDL is Occam's razor principle, where the simplest but well-interpreted model of known data should be used among all possible probability models. Thus, MDL criterion tend to choose simpler models that can describe the population well. ECGA algorithm can be divided into the following steps:

(1) Randomly initialize a population of N individuals.

(2) Choose S individuals with good fitness.

(3) Based on MDL criterion, search for MPM model with greedy search method.

(4) Stop the search when the MPM model converges.

(5) Generate a new generation of populations by sampling the MPM model.

(6) Return to 2 steps.

III. Proposed method

In this section, the proposed method is described in detail.

A. Algorithm Framework

Most of the traditional decomposition methods are offline, that is, grouping before optimization. The process of grouping consumes huge fitness evaluations, and the information in the grouping process can not be utilized by the optimization algorithm. In this paper, we propose a dynamic online grouping CC algorithm CC-MPM which uses the marginal distribution product model to describe population and to explore the hidden structure of the problem. The grouping process executes as long as the optimization process is considered as stagnant. Optimization and grouping are related, and they will be more adaptive for nonseparable functions and can maximize the utilization of computational resources .

Algorithm 1 show the framework of the proposed CC-MPM. Firstly, the original problem is decomposed into m n-dimensional sub-problems randomly (line 1). In this paper $n = 20$. Then the CC-MPM algorithm is initialized according to the grouping results. Each subproblem is optimized by a

Algorithm 1: Framework of CC-MPM

```
1  [group, FEused] ←random_grouping(Func);
2  [pop, fitness] ←Initial(Pop_size, Ubound, Lbound, group);
3  while FE < MaxFes do
4      for i = 1 to |group| do
5          [FE, pop{i}, best_eval] ←optimizer(group{i}, pop{i}, FE);
6          if best_eval unchanged then
7              stag_cnt ← stag_cnt + 1;
8          else
9              stag_cnt = 0;
10         end
11     end
12     if stag_cnt > 1000 then
13         [group, lnk]← mpm_grouping(pop, group, lnk);
14         [pop, fitness] ←Initial(Pop_size, Ubound, Lbound, group);
15     end
16 end
```

Algorithm 2: Variables grouping via MPM

```
Randomly select k variables from the original problem;
Discretize the variables;
Search for grouping structure with greedy search method;
Output the linkage in the grouping structure as lnk_fnd ;
Merge together lnk_fnd and linkage discovered
    previously as lnk ;
Random grouping variables not included in lnk, obtain
    grouping structure group;
Assign the variables in lnk to group, variables have
    linkage are assigned to the same subgroup;
Output group;
```

certain optimizer in round-robin fashion. Each subproblem only evolve one generation, and the current best solution and fitness value will be output (line 5). If the optimal fitness value $best\_eval$ has not updated for 1000 times in a row, the MPM grouping will be performed based on the current population (line 12-15). The termination condition of the whole algorithm is that it reaches a certain number function evaluations. After the algorithm is terminated, the resulting optimal solution and its function value will be output.

The main optimizer of CC-MPM is covariance matrix self-adaptation evolution strategy (CMA-ES) which had shown outstanding performances on continuous optimization problem. CMA-ES searches optimal solution by sampling and updating its parameters (step size, covariance, center) with sample information. In the proposed CC-MPM, the optimizer is a multiple population CMA-ES, which means that there are multiple CMA-ES optimizer in the search space simultaneously. The multiple populations schema is borrowed from literature [14].

B. Variable Grouping via MPM

The optimizer updates the global optimal value with the best value that has been found every generation. When the global optimal value has not been updated for 1000 consecutive times, the whole co-evolution process can be considered as stagnant which means fall into a certain local optimum. At this moment, we use the current existing population information to establish marginal product probability model, and calculate the MDL value of each model. Greedy search method is utilized to find the model with relatively small MDL value, and then the decision variables of the problem are regrouped according to the model found.

The MPM grouping method makes full use of the population information generated in optimization procedure. And the regrouping strategy is equivalent to the resumption of evolutionary process. Therefore, the algorithm is designed to trigger the regrouping when the co-evolution progresses is considered as stalling. Algorithm 2 show how to regroup decision variables according to current population.

The first step is to extract k decision variables, which is the target of this round of grouping, from the original problem. In this paper, $k = 30$. The reason why not learn the linkage of all the decision variables of the high-dimensional problem is that, as the number of variables increases, the number of potential models increases exponentially. It takes much time to do a search of the all grouping structure for all variables. Besides, we believe that the population information of only one generation may not fully reflect the linkage of all variables. The linkage need to be discovered dynamically with population information of different evolutionary stages. Therefore, the linkage learning is conduct in batches.

The second step is to discretize the variables. The original ECGA can only solve binary coded variables. In order to enable ECGA to solve continuous problems, we use Fixed-Width Histogram (FWH) [17] [18] technique to transform problems from the continuous domain into the discrete domain. For the domain interval $[l, u]$, FWH decompose the interval into k bins with equal $(u - l)/k$. The range of the ith bin is:

$$\left[l + \frac{i\,(u-l)}{k}, l + \frac{(i+1)\,(u-k)}{k}\right] \tag{1}$$

In this paper, $k = 5$, $i \in 0, 1, 2, 3, 4$. For example, if $X \in [-5, 5]$ and $x_1 = 4.23$, then x_1 will be coded as 4 in discrete domain.

Next step is to search a suitable marginal product model which is simple and can explain the current population well. The pseudocode of greedy search for MPM model is shown in Algorithm 3. Algorithm 3 can be divided into two phases: initialization (line 1-8), greedy search (line 9-24). In Algorithm 3, $group$ denote one grouping structure, $newgroups$ is a set contains many grouping structures of next generation, C_1, C_2 is the measurement of one grouping structure, $C^{'}$ is the normalized measurement of one grouping structure. For the convenience of expression, the $C^{'}$ of model is called the MDL value in this paper.

In this paper, we say that variables have linkage, if they are grouped into the same subgroup. After grouping, merge the linkage found lnk_fnd this time with linkage found previously, then get all the linkage found so far lnk. Then, variables have no linkage with any other variables are randomly grouped to several subgroups, each subgroup obtain 20 variables. The last step, assign variables in lnk to the subgroup of $group$, and output variable structure $group$. The reason why assign

Algorithm 3: Greedy search for model

$group \leftarrow$initialize model, one variable per subgroup;
$newgroups \leftarrow$ merge any two subgroups of $group$ to generate j new models;
for $i = 1$ *to* j **do**
  evaluate $newgroups\{i\}$, get $C_1(i)$ and $C_2(i)$;
end
$C^{'} \leftarrow$ normalize (C_1, C_2) ;
$C^{'}_{best} \leftarrow min(C^{'})$;
$best\_group \leftarrow newgroups$ with minimum $C^{'}$;
while *size*(*best_group*) > 1 **do**
  $newgroups \leftarrow$ merge any two subgroups of $best\_group$ to generate j new models;
  $newgroups \leftarrow newgroups \cup best\_group$;
  for $i = 1$ *to* j **do**
    evaluate $newgroup\{i\}$, get $C_1(i)$ and $C_2(i)$;
  end
  $C^{'} \leftarrow$ normalize (C_1, C_2) ;
  $C^{'}_{best\_new} \leftarrow min(C^{'})$;
  $best\_newgroup \leftarrow newgroups$ with minimum $C^{'}$;
  if $(C^{'}_{best\_new}) < C^{'}_{best}$ **then**
    $best\_group \leftarrow best\_newgroup$;
    $C^{'}_{best} \leftarrow C^{'}_{best\_new}$;
  else
    $break$;
  end
end
output $best\_group$;

lnk to $group$ is that the linkage we found is not complete. Assigning lnk to $group$ will get a higher grouping accuracy.

C. Normalization of MDL Measurement

According to the MDL criterion, MPM model is measured by the sum of model complexity C_1 (Eq. 2) and compressed population complexity C_2 (Eq. 3), that is, combined complexity ($C = C_1 + C_2$).

$$Model\ complexity = \log N \sum_I b^{S[I]} \tag{2}$$

$$Compressed\ population\ complexity = N \sum_I E(M_I) \tag{3}$$

$E(M_I)$ is the entropy of the marginal distribution of the subgroup I. N is the number of individual in population. $S[I]$ is the size of partition subset of an MPM. The number b depend on encoding mechanism, in this paper, domain interval is decomposed into 5 bins, therefore $b = 5$.

According to the MDL criterion, the MPM model searching refers to the search for a MPM model with minimal combined complexity in ECGA. In terms of large scale optimization, small adjustment need to be made in ECGA.

As more and more variables are combined into subgroup, the model complexity increases exponentially, and the compress population complexity will decrease. The extent of variation in these two quantities are disproportionate, especially when the number of variables is large. In large scale optimization, it is not appropriate to add two criterions directly to one, which leads to the failure of combining variables into subgroup. Therefore, the two criterions are normalized before added. Normalize model complexity and compressed population complexity of all models using equation:

$$C^{'}_m = \frac{C_m(g) - C_{m\_min}}{C_{m\_max} - C_{m\_min}} \tag{4}$$

where C_{m_min} and C_{m_max} is the minimum and maximum of complexity $(C_m, m = 1, 2)$ of one generation in greedy search, $C^{'}_m$ is the normalized complexity. After normalization, sum the two kinds of complexity $C^{'}_1$ and $C^{'}_2$ up, and get $C^{'}$.

IV. Experiments

In this section, we tested the proposed CC-MPM on CEC'10 LGSO benchmark function, and compared the results with other popular LSGO algorithms.

A. Experiment Setup

The CEC'10 LSGO benchmark consists of 20 1000-dimensional functions which fall into four categories:

1) *fully separable(F1-F3)*
2) *partially separable with one single 50-dimensional non-separable 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)*

Each algorithm had been performed 25 independent runs with maximum number of fitness evaluations (FEs) 3×10^6. Each subproblem evolves one generation in a single circle. The population number of CC-MPM optimizer is set to 10.

B. Performance Comparison

The proposed method is compared with other seven popular large scale CC algorithms which are summarized in Table I. The main optimizer of these seven algorithm is differential evolution, but the grouping strategy is different.

TABLE I
Description of compared algorithms.

Algorithm	Description
DECC-D [6]	Cooperative co-evolutionary differential evolution(DECC) with delta grouping
MLCC [19]	DECC with random grouping using a pool of potential subcomponent sizes
DECC-DML [6]	DECC with delta grouping using a pool of potential subcomponent sizes
DECC-DG [4]	DECC with differential grouping(DG)
DECC-I [4]	DECC with ideal grouping
CBCC-1 [20]	DECC-DG with contribution based CC algorithm
CBCC-2 [20]	DECC-DG with contribution based CC algorithm

Table II show the mean, standard deviation of the experimental results of 8 algorithms. The bold value indicate the best performance. As seen from Table II, algorithms with differential grouping, ideal grouping and MPM grouping get 18 out of 20 best results. The DECC-I has most of the best results, followed by the proposed CC-MPM with 5 best performance. Note that CC-MPM achieves the best results on fully nonseparable functions. Nonseparable functions are challenge to all CC algorithms, since any variable division structure is inappropriate. As an algorithm with dynamic online variables grouping method, CC-MPM can solve nonseparable problems better.

To verify the performance difference between CC-MPM and other seven algorithms in statistical sense, a pairwise Wilcoxon signed ranks test is conducted with 0.05 significant level. Table III show the detail information of the test results. '+', '-' and '$\sim$' denote the results of the proposed CC-MPM is significantly better, significantly worse and not significantly different with the compared algorithm, respectively. We can see that, CC-MPM have 44 out of 60 better results in comparison with DECC-D, MLCC, DECC-DML whose decomposition strategy is random grouping or delta grouping.

Compared with DG-based algorithm, CC-MPM has better results in 31 out of 60, and worse results in 28 out of 60. DG is a state-of-the-art decomposition strategy which achieve perfectly correct decomposition in twelve functions. So it is reasonable that the performance of CC-MPM is about tied with DG-based algorithms. Anyway, CC-MPM has the best performance on nonseparable functions.

V. CONCLUSIONS

In this study, a CC algorithm with dynamic online variable grouping is proposed. The grouping executes when the CC algorithm is identified as stagnant. The grouping method is inspired by the linkage learning strategy in ECGA which builds MPM models of current population and searches for a model with relatively small MDL value. A discretization technique of decision variables and normalization method of MDL value is used to make the linkage learning strategy more adaptive in large-scale continuous optimization.

The proposed method is tested on the CEC'10 LSGO benchmark. The result show that, the proposed CC-MPM outperformed three CC algorithm whose decomposition strategy is random grouping and delta grouping. CC-MPM is matched in performance with DG-based CC algorithm. Furthermore, CC-MPM has gained the best performance on nonseparable functions which verified the idea that dynamic online grouping strategy is more adaptive for nonseparable problems.

The number of stagnant generations trigger grouping is set to 1000 in this paper. This parameter can affect the performance of the proposed method. Therefore, further study is needed to find a more adaptive setting of this parameter.

ACKNOWLEDGMENT

This work has been supported by National Science Foundation of China(No.6117235, 61473233).

REFERENCES

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, p. 504, 2006.

[2] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July*, 2011, pp. 265–272.

[3] M. A. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization," *Lecture Notes in Computer Science*, vol. 866, pp. 249–257, 1994.

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

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

[6] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *IEEE Congress on Evolutionary Computation*, 2011, pp. 1–8.

[7] G. R. Harik, F. G. Lobo, and K. Sastry, *Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA)*, 2006.

[8] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions," *Biosystems*, vol. 39, no. 96, pp. 263–278, 1960.

[9] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 42, no. 2, p. 13, 2016.

[10] Y. Sun, M. Kirley, and S. K. Halgamuge, "A recursive decomposition method for large scale continuous optimization," *IEEE Transactions on Evolutionary Computation*, 2017.

[11] M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, "Dg2: A faster and more accurate differential grouping for large-scale black-box optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 929–942, 2017.

[12] L. Panait, S. Luke, and R. P. Wiegand, "Biasing coevolutionary search for optimal multiagent behaviors," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 629–645, 2006.

[13] X. Peng and Y. Wu, "Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering," *Swarm and Evolutionary Computation*, 2017.

[14] ——, "Large-scale cooperative co-evolution with bi-objective selection based imbalanced multi-modal optimization," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 1527–1532.

[15] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *Evolutionary Computation IEEE Transactions on*, vol. 3, no. 4, pp. 287–297, 1999.

[16] J. Rissanen, "Stochastic complexity in statistical inquiry," *World Scientific*, vol. 15, p. 188, 1989.

[17] Y. P. Chen and C. H. Chen, "Enabling the extended compact genetic algorithm for real-parameter optimization by using adaptive discretization," *Evolutionary Computation*, vol. 18, no. 2, p. 199, 2010.

[18] S. Tsutsui, M. Pelikan, and D. E. Goldberg, "Evolutionary algorithm using marginal histogram models in continuous domain," in *Continuous Domain, Proc of the Genetic and Evolutionary Computation Conference Workshop Program*, 2001, pp. 230–233.

[19] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 1663–1670.

[20] M. N. Omidvar, X. Li, and X. Yao, "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms," in *Conference on Genetic and Evolutionary Computation*, 2011, pp. 1115–1122.

TABLE II

PERFORMANCE COMPARISON ON CEC'10 LSGO BENCHMARK FUNCTIONS.

Funtions		DECC-D	MLCC	DECC -DML	DECC-DG	DECC-I	CBCC-1	CBCC-2	CC-MPM
F1	Mean	**0.00E+00**	8.24E-07	2.77E-07	2.08E+06	3.83E+05	1.96E+06	6.38E+06	7.74E+05
	Std	0.00E+00	4.28E-7	9.60E-07	2.05E+06	6.51E+05	1.99E+09	1.82E+07	3.98E+05
F2	Mean	6.52E+01	**2.61E-03**	1.04E+01	4.22E+03	4.39E+03	4.33E+03	4.18E+03	1.58E+03
	Std	4.47E+01	5.34E-03	2.24E+01	3.80E+02	2.96E+02	3.04E+02	5.38E+02	1.38E+02
F3	Mean	2.29E+00	1.27E-02	2.57E-01	1.09E+01	1.10E+01	1.12E+01	1.10E+01	**0.00E+00**
	Std	1.75E-01	2.65E-02	7.06E-01	8.53E-01	6.23E-01	8.96E-01	7.32E-01	0.00E+00
F4	Mean	2.98E+12	1.17E+14	1.18E+14	5.06E+11	2.71E+10	1.81E+11	**1.65E+10**	7.80E+11
	Std	9.35E+11	4.12E+13	1.69E+14	1.96E+11	1.24E+10	1.08E+11	3.62E+09	2.43E+11
F5	Mean	2.86E+08	5.04E+08	4.99E+08	7.36E+07	6.86E+07	7.02E+07	**6.43E+07**	3.76E+08
	Std	1.08E+08	1.36E+08	1.28E+08	9.56E+06	1.24E+07	1.05E+07	1.31E+07	5.53E+07
F6	Mean	5.89E+06	1.90E+07	1.68E+07	**1.58E+01**	1.63E+01	8.14E+04	4.11E+04	1.86E+02
	Std	5.43E+06	2.12E+06	6.08E+06	7.30E-01	9.69E-01	2.84E+05	2.05E+05	1.75E+02
F7	Mean	1.47E+05	4.88E+10	3.42E+10	2.79E+04	**1.17E+04**	1.23E+05	1.26E+10	1.10E+07
	Std	2.47E+05	1.64E+10	5.19E+10	2.03E+04	3.96E+03	1.09E+05	1.48E+10	8.25E+06
F8	Mean	1.27E+08	8.23E+08	3.10E+10	2.78E+07	**8.06E+05**	7.50E+06	3.72E+07	1.00E+08
	Std	1.52E+08	1.92E+08	6.90E+10	3.19E+07	1.63E+06	1.84E+07	3.47E+07	3.18E+07
F9	Mean	1.01E+08	1.69E+09	1.05E+09	3.65E+07	4.76E+07	**1.02E+07**	3.40E+08	1.93E+07
	Std	9.09E+06	2.54E+08	1.13E+09	1.49E+07	5.30E+07	3.84E+06	2.67E+08	1.12E+07
F10	Mean	4.07E+03	5.19E+03	4.30E+03	3.33E+03	3.13E+03	**2.59E+03**	4.90E+03	3.54E+03
	Std	1.26E+03	1.72E+03	1.77E+03	1.92E+02	1.68E+02	1.48E+02	6.37E+02	1.32E+02
F11	Mean	9.98E+01	2.00E+02	1.91E+02	2.64E+01	**2.51E+01**	2.69E+01	2.75E+01	1.27E+02
	Std	1.01E+02	2.24E+00	3.56E+01	2.95E+00	2.72E+00	2.64E+00	3.18E+00	1.78E+01
F12	Mean	9.14E+03	8.68E+05	4.76E+05	3.21E+04	2.44E+04	3.53E+04	5.07E+04	**5.67E+00**
	Std	1.08E+03	1.24E+05	4.69E+05	1.06E+04	7.12E+03	1.11E+04	1.10E+04	5.25E+00
F13	Mean	5.44E+03	3.24E+04	8.62E+04	2.89E+07	**1.29E+04**	9.06E+04	1.29E+07	1.72E+05
	Std	2.76E+03	2.61E+04	1.95E+05	1.57E+07	4.34E+03	6.11E+04	7.36E+06	3.48E+05
F14	Mean	3.00E+08	3.62E+09	2.22E+09	2.10E+07	2.14E+07	2.24E+07	5.35E+09	**9.89E+06**
	Std	2.19E+07	5.43E+08	2.04E+09	2.25E+06	2.06E+06	2.27E+06	6.00E+08	1.06E+06
F15	Mean	1.30E+04	1.17E+04	1.10E+04	**2.88E+03**	2.84E+03	2.84E+03	3.22E+03	4.76E+03
	Std	2.18E+02	2.05E+03	2.77E+03	2.76E+02	1.86E+02	2.65E+02	4.17E+02	2.46E+02
F16	Mean	2.02E+02	3.99E+02	3.62E+02	1.97E+01	1.93E+01	1.87E+01	**1.91E+01**	2.89E+02
	Std	1.58E+02	3.43E+00	1.09E+02	3.61E+00	3.77E+00	3.83E+00	2.76E+00	2.19E+01
F17	Mean	7.47E+04	1.79E+06	9.71E+05	7.76E+00	**7.08E+00**	1.49E+01	1.24E+02	3.29E+01
	Std	4.72E+03	1.78E+05	1.05E+06	1.89E+00	1.76E+00	7.01E+00	5.72E+01	1.47E+01
F18	Mean	1.44E+04	1.07E+05	7.77E+04	2.01E+10	**1.15E+03**	4.10E+09	1.23E+11	1.97E+03
	Std	1.27E+04	2.68E+04	1.75E+05	4.82E+09	1.65E+02	1.83E+09	1.45E+10	5.14E+02
F19	Mean	1.59E+06	2.96E+06	2.70E+06	9.01E+05	8.95E+05	9.12E+05	9.11E+05	**2.95E+05**
	Std	1.32E+06	4.29E+05	3.37E+06	6.14E+04	6.24E+04	7.11E+04	6.02E+04	3.79E+04
F20	Mean	2.27E+03	1.75E+05	5.42E+03	6.53E+08	1.67E+07	1.41E+07	6.97E+09	**1.06E+03**
	Std	2.44E+02	2.08E+05	1.46E+04	6.71E+08	3.30E+07	1.96E+07	1.12E+09	8.31E+01
Best No.		1	1	0	2	6	2	3	5

TABLE III

WILCOXON TEST WITH A SIGNIFICANT LEVEL OF 0.05.'+"-',AND '∼' DENOTE SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE AND 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	+	+	+	-	-	-	+	-	+	+	-	+	+	+	-	-	+	+	+	+