Smart Use of Computational Resources Based on Contribution for Cooperative Co-evolutionary Algorithms

Mohammad N. Omidvar
School of Computer Science
and Information Technology
RMIT University
Melbourne, Australia
momidvar@cs.rmit.edu.au

Xiaodong Li
School of Computer Science
and Information Technology
RMIT University
Melbourne, Australia
xiaodong.li@rmit.edu.au

Xin Yao
School of Computer Science
The University of Birmingham
Birmingham
United Kingdom
xin@cs.bham.ac.uk

ABSTRACT

Standard Cooperative Co-evolution uses a round-robin method to select subcomponents to undergo optimization. In a non-separable (epistatic) optimization problem, dividing the computational budget equally between all of the subcomponents is not necessarily the best strategy. When dealing with non-separable problems, there is usually an imbalance between the contribution of various subcomponents to the global fitness of the individuals. Using a round-robin fashion treats all of the subcomponents equally and wastes the computational budget. In this paper, we propose a Contribution Based Cooperative Co-evolution (CBCC) that selects the subcomponents based on their contributions to the global fitness. This alleviates the imbalance issue and allows the computational resources to be used more efficiently. Experiments on several benchmark functions with the "imbalance issue" show that this new scheme is promising, especially when it is combined with a grouping algorithm that captures interacting variables in common subcomponents.

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems; G.1.6 [**Mathematics of Computing**]: Optimization—*Global Optimization*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Performance

Keywords

contribution based cooperative co-evolution, large scale optimization, non-separability, evolutionary algorithms

1. INTRODUCTION

Evolutionary Algorithms (EAs) have been very successful in solving many optimization problems [4, 1], however their performance

deteriorates as the dimensionality of the problem increases. *Curse of dimensionality* [2] refers to the exponential growth in the size of the search space as the dimensionality increases. This makes large scale optimization a very challenging task. Cooperative Co-evolution (CC) has been proposed by Potter and De Jong in order to bring a notion of problem decomposition into EAs [10]. This scheme is very promising for large scale optimization because of its potential in decomposing a high dimensional problem into a set of lower dimensional problems.

Various decomposition strategies have been suggested. In the original CC implementation, the decision vector is decomposed into a set of low-dimensional problems. In other words, every subcomponent evolves only one of the decision variables at a time [10]. In a more recent work [13] the decision variables are divided into k s-dimensional subcomponents each of which is evolved separately in a round-robin fashion (where $k \times s$ equals to the total number of dimensions n). Since an individual in one subcomponent does not form a complete solution, it will be concatenated by selected individuals from other subcomponents (usually best members of other subcomponents) before evaluation. This is where the cooperation happens. These static decomposition strategies are only effective when the interactions of variables between subcomponents are kept at minimum [6]. This interdependency between the decision variables is commonly referred to as *non-separability* [11] or *epistasis* [5]. As remarked in [7]: "*In other words, epistasis measures the extent to which the contribution to fitness of one gene depends on the values of other genes*". Salomon has shown that the performance of EAs degrades significantly in optimizing non-separable problems [11]. Without variable interaction, a large scale problem could be decomposed into a set of one dimensional problems each of which is optimized for a predefined number of iterations in a round-robin fashion. However, this is not an effective decomposition under variable interaction. In order to benefit from a CC framework in a non-separable domain, we need to identify and group the interacting variables in common subcomponents.

Random Grouping [14] and *Delta Grouping* [9] are two major attempts in capturing interacting variables in common subcomponents. Cooperative Co-evolution with Variable Interaction Learning (CCVIL) [3], is another grouping technique recently proposed. Delta grouping in particular has shown great performance where there is only one group of interacting variables [9]. By the advent of such grouping techniques, it is now possible to quantify the contribution of the separable and non-separable portions of an optimization problem to the fitness of the individuals. In this paper we demonstrate that there is usually an imbalance between the contribution of the non-separable and the separable portions of an optimization problem to the global fitness. We show that the round-

robin strategy of CC in selecting a subcomponent to undergo optimization is very ineffective where there is an imbalance between the separable and non-separable portions of the fitness value. In many cases the value of the separable portion of the fitness is negligible compared to its non-separable portion (or vice versa). This means that even if the separable portion is optimized to its optimum value, it will have minimum effect on the global fitness. In such scenarios, most of the effort needs to be focused on the subcomponents with the most significant effect on the global fitness. Round-robin strategy fails in such situations by putting equal emphasis on all of the subcomponents regardless of their fitness share, wasting considerable amount of fitness evaluations (FEs).

In order to mitigate this deficiency of round-robin strategy, we proposed a Contribution Based Cooperative Co-evolution (CBCC) where the subcomponents are selected for further optimization based on their contributions towards improving the global fitness. In a nutshell, the rationale behind this new contribution based cooperative co-evolution is to reduce the imbalance in the contributions to the global fitness between the separable and non-separable components as much as possible. Given a grouping strategy that can identify non-separable and separable subcomponents, this technique keeps the separable and non-separable components of the fitness value as close as possible to each other. Failing to do so will result in a significant imbalance in subcomponent's contribution to the global fitness. In such situations the true improvement in some of the subcomponents – no matter how well they converged – will not be reflected in the global fitness simply because they are negligible as compared to the fitness of other subcomponents. Any effort in further improving the individuals in such subcomponents will be simply wasted in the co-evolutionary process. The imbalance issue and the mechanics of the CBCC algorithm is further described in Sections 2 and 4 respectively. The organization of the rest of this paper is as follows. Section 2 describes the imbalance issue in more details. Section 3 covers the preliminaries. Section 4 describes the CBCC algorithm. The experimental results and the analysis of the empirical studies are described in Sections 5 and 6 respectively. Finally, Section 7 concludes the findings of the paper and sketches possible future investigations.

2. THE IMBALANCE PROBLEM

It was described earlier in Section 1 that in some problems, there may be an imbalance between the fitness contribution of the non-separable and separable subcomponents to the global fitness. In Cooperative Co-evolution, a round-robin method is employed to optimize all of the subcomponents in an iterative manner. This switching strategy splits the computational budget *equally* between all of the subcomponents. However, it is clear that in the presence of an imbalance between the fitness contribution of various subcomponents, this strategy is ineffective.

Here we use a concrete example to illustrate the imbalance issue between the subcomponents. For our discussion we selected f_4 from CEC'2010 benchmark suite [12] as a representative which is defined as follows:

$$
\begin{aligned}
f_4(\vec{x}) \;=\; & 10^6 \times f_{elliptic}(R[x_{p_1}, ..., x_{p_m}]^T) + \\
& f_{elliptic}([x_{p_{m+1}}, ..., x_{p_n}]^T) \;,
\end{aligned} \tag{1}
$$

where n is the dimensionality of the problem, m is the dimensionality of the non-separable portion, R is a rotation matrix to create variable interaction, and $\vec{p}$ is a random permutation vector with values from $\{1, ..., n\}$. The variables m and n are set to 50 and 1000 respectively according to [12]. So f_4 contains a 50-dimensional non-separable subcomponent based on rotated elliptic function and

a 950-dimensional separable subcomponent based on the shifted elliptic function ($f_{elliptic}$). Note that the coefficient 10^6 in Equation (1) creates the imbalance between the two components, giving the non-separable component more weight in the overall fitness.

Our investigations on f_4 have shown that its initial average fitness of 25 independent runs is approximately 5.16×10^{15}. In order to quantify the fitness contribution of separable and non-separable portions of the fitness we need to record them separately. The newly proposed CEC'2010 benchmark functions gives us this flexibility. Note that our proposed CBCC algorithm solely relies on the global fitness value to estimate the contribution of various subcomponents (See Section 4).

Recording the separable and non-separable subcomponents of the fitness function revealed that the initial fitness of the non-separable subcomponent is approximately 5.16×10^{15} while the fitness of the separable subcomponent is approximately 3.44×10^{11}. Note that the global fitness is the sum of these two components. It is clear that the contribution of the separable subcomponent in the global fitness is negligible as compared to the non-separable subcomponent. It is obvious that any effort in further optimizing the separable portion will not improve the global fitness significantly and any major improvement is due to improvements in the non-separable subcomponent. The round-robin strategy in CC treats both of the subcomponents equally and in the case of f_4, approximately half of the fitness evaluations would be wasted. On the other hand, CBCC algorithm uses these information to split the computational budget between various subcomponents based on their contributions. It is notable that CBCC works well as long as there is an imbalance between various subcomponents, no matter which subcomponent has the maximum contribution. For example in the case of f_4, if the separable portion has a greater contribution in the global fitness, any effort in optimizing the non-separable portion would be waste of computational budget.

Without any systematic decomposition of decision variables into separable and non-separable subcomponents, round-robin strategy may be the best possible solution for choosing subcomponents to undergo optimization. However, using more systematic grouping techniques such as delta grouping [9] and CCVIL [3], it is now possible to identify non-separable variables from the separable ones. Being able to quantify the fitness contribution of each of the subcomponents, more computational budget can be spent on the subcomponents with the greatest contribution in the improvement of the global fitness value. Clearly, decomposition strategies (which we will describe in the following section) play an important role in determining the contributions of the subcomponents.

3. GROUPING STRATEGIES

In this section we describe the two decomposition strategies that we used in our experiments, namely delta grouping and ideal grouping.

3.1 Delta Grouping

Delta grouping is a systematic approach for capturing interacting variables [9]. According to Salomon [11], the improvement interval of variables shrink considerably when dealing with non-separable problems. The rationale behind the delta grouping is that when two interacting variables are grouped in separate subcomponents, there will be a limit to the extent these variables could be optimized towards their global optimum [9]. Delta grouping is based on this idea and sorts the decision variables based on their absolute magnitude of change in each of the dimensions. In other words, a delta vector is formed based on the absolute difference in the centroid of populations in two consecutive iterations. By sorting the decision

variables based on their corresponding delta values, it will be more likely to place interacting variables with similar delta values close to each other. Finally, in a CC framework the grouping of variables is determined based on the sorted delta values.

3.2 Ideal Grouping

Ideal grouping refers to the decomposition of decision variables into a set of subcomponents where there is absolutely no interdependency between any two subcomponents. In order to perform an unbiased analysis of CBCC algorithm, we decided to perform a manual ideal grouping of the decision variables based on the test functions provided in the CEC'2010 benchmark functions [12] since we know exactly which variables interact with each other. The purpose of using this "ideal grouping" information is to facilitate the comparison of our proposed CBCC model in the context of such ideal situation in order to determine the effectiveness of CBCC. For the sake of completeness, the performance of CBCC is also benchmarked in combination with one of the systematic grouping strategies. We have chosen delta grouping for our studies. We believe that the performance of grouping strategies will continue to improve in near future, making an ideal decomposition of the decision variables a feasible and achievable goal. For example, the recently proposed CCVIL [3] seems to suggest that it is possible to obtain a near optimal grouping of the decision variables.

4. CONTRIBUTION BASED COOPERATIVE CO-EVOLUTION

Contribution Based Cooperative Co-evolution attempts to divide the computational budget based on the contribution of different subcomponents towards improving the global fitness. It was mentioned in previous section that the CBCC algorithm only relies on the global fitness to calculate the contribution of separable and non-separable subcomponents of the fitness function. This is not possible without any systematic grouping of the decision variables in order to minimize the interdependency between the subcomponents.

Under ideal decomposition, there is no dependency between the subcomponents. So when one of the subcomponents undergoes optimization while other subcomponents are kept intact, the changes in the global fitness value will be the reflection of the improvement in the selected subcomponent. Under a suboptimal decomposition such as delta grouping, such calculations will be less accurate. However, delta grouping still serves as a good approximation to calculate the contributions of various subcomponents. It is clear that the accuracy of the measurements relies on the accuracy of decomposition strategy. According to [9], delta grouping is capable of capturing interacting variables with reasonable accuracy when the fitness function has only one non-separable component.

In this paper, we propose the CBCC algorithm to estimate the contribution of each of the subcomponents by measuring the changes in the global fitness when the respective subcomponent undergoes optimization. Algorithm 1 shows how CBCC uses these information to select various subcomponents for optimization. CBCC is very similar to a conventional CC. After the initialization, each of the subcomponents which are formed using either delta or ideal grouping will be selected in a round-robin fashion to measure their initial contributions in the global fitness (Algorithm 1, lines 12-19). We call this the *testing phase*. It can be seen that the array ΔF keeps track of the changes in the fitness for all of the subcomponents. In the next stage (Algorithm 1, lines 20-30) the subcomponent with the greatest entry in ΔF is selected for further optimization. The difference between the two versions of the CBCC (i.e. CBCC1 and CBCC2) is in this stage. In CBCC1, the selected sub-

---

Algorithm 1: CBCC($cycles, FEs, version$)

1. $pop[1:N_{popsize}, 1:n] \leftarrow random\ population$
2. $(best, best\_val) \leftarrow min(evaluate(pop))$
3. $oldpop \leftarrow pop$
4. $prev\_best\_val \leftarrow best\_val$
5. **for** $i \leftarrow 1$ **to** $cycles$ **do**
6.   **if** $grouping_method = $ "ideal" **then**
7. $(groups, num\_groups) \leftarrow ideal\_grouping()$
8.   **else**
9. $(groups, num\_groups) \leftarrow delta\_grouping()$
10.   **if** $i = 1$ **then**
11. $\Delta F \leftarrow zeros(1, num\_groups)$
12.   **for** $j \leftarrow 1$ **to** num_groups **do**
13. $indicies \leftarrow groups(j)$
14. $subpop \leftarrow pop[:, indicies]$
15. $subpop \leftarrow SaNSDE(best, subpop, FEs)$
16. $pop[:, indicies] \leftarrow subpop$
17. $prev\_best\_val \leftarrow best\_val$
18. $(best, best\_val) \leftarrow min(evaluate(pop))$
19. $\Delta F[i] \leftarrow \Delta F[i] + prev\_best\_val - bestval$
20. $\delta \leftarrow 1$
21.   **while** $\delta \neq 0$ **do**
22. $\Delta F\_indicies \leftarrow sort\_descending(\Delta F)$
23. $indicies \leftarrow groups(\Delta F\_indicies[1])$
24. $subpop \leftarrow pop[:, indicies]$
25. $subpop \leftarrow SaNSDE(best, subpop, FEs)$
26. $pop[:, indicies] \leftarrow subpop$
27. $prev\_best\_val \leftarrow best\_val$
28. $(best, best\_val) \leftarrow min(evaluate(pop))$
29. $\delta \leftarrow prev\_best\_val - bestval$
30. $\Delta F[\Delta F\_indicies[1]] \leftarrow \Delta F[\Delta F\_indicies[1]] + \delta$
31.   **if** $version = $ "CBCC1" **then**
32. $\delta \leftarrow 0$

---

component is optimized for only one iteration whereas in CBCC2, the selected subcomponent will be optimized as long as it improves the fitness. When no improvement is identified in the selected subcomponent, the algorithm enters the testing phase to give all of the subcomponents another chance to update their contributions in ΔF. Note that the contribution information is accumulated from the first cycle.

5. EXPERIMENT SETUP

We conducted our experiments based on the newly proposed CEC'2010 benchmark functions which have been proposed for the Special Session and Competition on Large Scale Global Optimization in CEC'2010 [12]. The advantage of this benchmark suite is that the degree of non-separability is well defined and is adjustable. This set of benchmark functions is scalable and contains five major categories which are as follows:

1. **Separable Functions** (f_1-f_3)

2. **Single-group m-nonseparable Functions** (f_4-f_8)
   f_4: Single-group Shifted and m-rotated Elliptic Function.
   f_5: Single-group Shifted and m-rotated Rastrigin's Function.
   f_6: Single-group Shifted and m-rotated Ackley's Function.
   f_7: Single-group Shifted and m-rotated Schwefel's Problem 1.2.
   f_8: Single-group Shifted and m-rotated Rosenbrock's Function.

3. $\frac{n}{2m}$**-group m-nonseparable Functions** (f_9-f_{13})

4. $\frac{n}{m}$**-group m-nonseparable Functions** (f_{14}-f_{18})

5. **Nonseparable Functions** (f_{19}-f_{20})

In the above list, m refers to the number of interacting variables in a non-separable group and n is the dimensionality of the problem. We have used the original values used in [12] which are 50 and 1000 respectively. We have excluded the fully separable functions (f_1-f_3) and fully non-separable functions (f_{19}-f_{20}) from our experiments because the focus of this paper is on problems where there are a mixture of separable and non-separable variables. From the remaining functions we focused our analysis more on functions

Table 1: Result of CBCC variants and DECC over 1000 dimensions using delta grouping (averaged over 25 runs).

Functions	CBCC1-D	CBCC2-D	DECC-D
f_4	3.884424e+12	3.953996e+12	3.994117e+12
f_5	3.846258e+08	3.934090e+08	4.162337e+08
f_6	4.025218e+06	2.449806e+06	1.356873e+07
f_7	4.214883e+07	4.222071e+07	6.578934e+07
f_8	3.604591e+07	5.639487e+07	5.392069e+07
f_9	6.240636e+07	6.231822e+07	6.187354e+07
f_{10}	1.181007e+04	1.193022e+04	1.156625e+04
f_{11}	5.611686e+01	5.592898e+01	4.764118e+01
f_{12}	1.538192e+05	1.512779e+05	1.527193e+05
f_{13}	9.168733e+02	1.066448e+03	9.867780e+02
f_{14}	1.981534e+08	2.040980e+08	1.983536e+08
f_{15}	1.534865e+04	1.542186e+04	1.531490e+04
f_{16}	1.525727e+02	1.698745e+02	1.880495e+02
f_{17}	8.994593e+05	8.893276e+05	9.030164e+05
f_{18}	2.891205e+03	2.366799e+03	2.123339e+03

Table 2: Result of CBCC variants and DECC over 1000 dimensions using ideal grouping (averaged over 25 runs).

Functions	CBCC1-I	CBCC2-I	DECC-I	MA-SW-Chains [8]
f_4	5.1828e+11	4.9948e+11	6.1306e+11	3.53e+11
f_5	1.1192e+08	1.1468e+08	1.3439e+08	1.68e+08
f_6	1.6331e+01	1.6301e+01	1.6358e+01	8.14e+04
f_7	5.8431e+02	9.6063e+02	2.9676e+01	1.03e+02
f_8	1.0907e+02	4.7965e+05	3.1908e+05	1.41e+07
f_9	5.5926e+07	6.2658e+07	4.8397e+07	1.41e+07
f_{10}	3.9186e+03	3.9134e+03	4.3361e+03	2.07e+03
f_{11}	1.0082e+01	1.0392e+01	1.0206e+01	3.80e+01
f_{12}	2.3631e+03	2.3664e+03	1.4754e+03	3.62e-06
f_{13}	1.1129e+03	1.1050e+03	7.5150e+02	1.25e+03
f_{14}	3.6539e+08	3.5944e+08	3.3762e+08	3.11e+07
f_{15}	5.8581e+03	5.8682e+03	5.8754e+03	2.74e+03
f_{16}	6.0027e-13	8.2380e-13	2.4741e-13	9.98e+01
f_{17}	4.5081e+04	4.5663e+04	3.9151e+04	1.24e+00
f_{18}	1.1543e+03	1.1328e+03	1.1694e+03	1.30e+03

f_4 to f_8 due to their imbalance characteristic. The imbalance issue is less significant in functions f_9 to f_{13} and functions f_{14} to f_{18} do not have any imbalance issue.

For every function, we run each of the algorithms for 25 independent runs and the mean and median are recorded. The dimensionality of all of the functions is set to 1000. We used SaNSDE [15] as the subcomponent optimizer in a CC framework and its population size of the subcomponent optimizers is set to 50. We call this algorithm DECC. The maximum number of fitness evaluations is set to 3×10^6 according to [12].

6. ANALYSIS OF RESULTS

This section contains the experimental results and the analysis of the outputs. Table 1[1] contains the average fitness of the 25 independent runs using delta grouping. The results for two variants of CBCC and the standard round-robin implementation of CC (DECC) are recorded in this table. Table 2 contains similar information but ideal grouping is used instead of delta grouping.

It can be seen from Table 1 that both CBCC variants outperform DECC on the second category of the benchmark functions (f_4-f_8) with the exception of f_8 where only CBCC-1 outperforms DECC. The reason that we don't see the same trend on the next category (f_9-f_{13}) might be due to poor performance of delta grouping in dealing with functions with more than one non-separable components. Although there is still a separable subcomponent in this category of functions, the poor performance of delta grouping affects

---

[1] The last character in CBCC1-D or DECC-D refers the grouping strategy. 'D' stands for delta and 'I' stands for ideal grouping.

the performance of the contribution based strategy in choosing the right subcomponents for optimization. For the next category of functions (f_{14}-f_{18}), since all of the components of the benchmark functions are non-separable, we do not expect to see any advantage in CBCC. This is because there is no imbalance between the fitness share of various components and hence all of the subcomponents should have equal chance of being selected for optimization. As a result, there is no clear trend for functions f_{14}-f_{18} in Table 1 and in most cases the performance of CBCC and DECC is similar.

As we mentioned earlier in Section 3.2, although delta grouping is effective in capturing the interacting variables for the second category of the benchmark functions, it is still an approximation technique which is far from an ideal grouping. The real contribution of a subcomponent is reflected only when the dependency between the subcomponents is kept to a minimum. This happens when we have an ideal separation of non-separable variables. So in order to better analyze the effect of the new contribution based CC, we also report the performance of both variants of CBCC and standard round-robin CC using ideal grouping of the decision variables. Table 2 contains similar results to Table 1, however ideal grouping is used instead of the delta grouping.

Table 2 shows the same trend as Table 1 except for f_7 and f_8. In order to justify this behavior we need to study the fitness of the separable and non-separable components of the fitness function separately, which is further described later in this section (see Tables 3 and 4). We can also see from Table 2 that on functions f_9-f_{13}, although the imbalance issue is less significant, at least one variant of CBCC has a better performance than its round-robin counterpart (DECC). Functions f_{10} and f_{11} behave as such. Note that this is hardly the case in delta grouping due to its poor performance on this category of functions (Table 1). We also compared the results in Table 2 with MA-SW-Chains [8], which is ranked first in CEC'2010 Competition on Large Scale Global Optimization [12]. It is interesting to see that both CBCC versions outperformed MA-SW-Chains on f_5, f_6 and f_8 from the second category of functions and on f_{11}, f_{13}, f_{16}, and f_{18} on the rest of the categories.

Looking at the global fitness value in Tables 1 and 2 is not sufficient for proper understanding CBCC's behavior. So in order to better understand the effect of CBCC, we recorded the fitness value for the separable and the non-separable portions of the fitness function separately in Table 3. It can be seen from the table that the result of Wilcoxon test is provided for significance testing. Note that the test is conducted for the separable and non-separable subcomponents separately. The research hypothesis is as follows: H_r: CBCC < DECC. In addition to these information the number of times that the separable and non-separable subcomponents are selected for optimization is also reported in Table 3. Since there are 19 non-separable subcomponents in CBCC-D, the grouping information for the non-separable subcomponent is based on the subcomponent that captured the maximum number of interacting variables. The number of times that this particular subcomponent is selected to be optimized, is also based on the average of the 25 runs.

By looking at Table 3, we can see that both variants of CBCC spend more time on the subcomponent that contains most of the interacting variables and hence less time is spent on the separable component. This shows that CBCC picks the subcomponent that has the maximum contribution in the improvement of the global fitness. By looking at the fitness values for the non-separable component of the fitness, we can see that in most cases, the value of the CBCC variants is lower than the corresponding value of their round-robin counterpart (DECC-D). This is intuitive as the contribution based methods spend more iterations on non-separable sub-

Table 3: Empirical results of CBCC and DECC on f_4 to f_8 using delta grouping with separate fitness values for non-separable and separable subcomponents. The frequency that each subcomponent is selected for optimization is also recorded.

Delta Grouping $n = 1000$		DECC-D		CBCC1-D		CBCC2-D		CBCC1-D *vs* DECC-D	
		Non-sep	Sep	Non-sep	Sep	Non-sep	Sep	Non-sep	Sep
f_4	Mean	4.192954e+12	2.815573e+7	3.884384e+12	4.02947e+7	3.953953e+12	4.423631e+7		
	Median	3.804919e+12	2.633791e+7	3.499598e+12	3.278075e+7	3.999183e+12	4.197518e+7	0.2305	0.9245
	Times optimized	3000	3000	3773	2959	4003	2947		
f_5	Mean	4.396958e+8	2.282362e+2	3.846256e+8	2.22091e+2	3.934088e+8	2.172886e+2		
	Median	4.447408e+8	2.264145e+2	3.950831e+8	2.230814e+2	3.651457e+8	2.200173e+2	0.0199	0.1758
	Times optimized	3000	3000	3431	2977	3545	2971		
f_6	Mean	1.589092e+6	1.568911e+0	4.025214e+6	4.05105e+0	2.449804e+6	2.497631e+0		
	Median	7.105427e-9	1.98952e-13	7.105427e-9	2.096101e-13	7.105427e-9	2.167155e-13	0.6424	0.9972
	Times optimized	3000	3000	4111	2941	4819	2904		
f_7	Mean	4.951645e+7	4.770511e+5	4.156518e+7	5.836553e+5	4.164323e+7	5.77486e+5		
	Median	4.643947e+7	4.790678e+5	4.046763e+7	5.86592e+5	4.034548e+7	5.890547e+5	0.0627	1.0000
	Times optimized	3000	3000	4458	2923	5097	2899		
f_8	Mean	4.132078e+7	3.079968e+0	3.60471e+7	4.240595e+0	1.287568e+9	6.29461e+1		
	Median	3.053601e+7	6.576564e-1	2.553446e+7	7.968974e-1	6.013773e+7	3.697347e-1	0.2803	0.7515
	Times optimized	3000	3000	3200	2989	3860	2959		

Table 4: Empirical results of CBCC and DECC on f_4 to f_8 using ideal grouping with separate fitness values for non-separable and separable subcomponents. The frequency that each subcomponent is selected for optimization is also recorded.

Ideal Grouping $n = 1000$		DECC-I		CBCC1-I		CBCC2-I		CBCC1-I *vs* DECC-I	
		Non-sep	Sep	Non-sep	Sep	Non-sep	Sep	Non-sep	Sep
f_4	Mean	6.130147e+11	2.332101e+4	5.182743e+11	2.067237e+6	4.99486e+11	3.475966e+5		
	Median	5.822549e+11	3.629365e+3	5.551435e+11	2.839523e+5	4.85171e+11	2.260827e+5	0.0935	1.0000
	Times optimized	30000	30000	39998.6	20001.4	40447.04	19554.16		
f_5	Mean	1.343821e+8	4.090196e+3	1.119111e+8	4.058337e+3	1.146796e+8	4.130307e+3		
	Median	1.367885e+8	4.064429e+3	1.146625e+8	4.048534e+3	1.159609e+8	4.115858e+3	0.0006	0.3708
	Times optimized	30000	30000	39999.48	20000.52	40022.4	19978.6		
f_6	Mean	3.552714e-9	1.635790e+1	3.552714e-9	1.633141e+1	3.552714e-09	1.630051e+1		
	Median	3.552714e-9	1.634814e+1	3.552714e-9	1.617864e+1	3.552714e-9	1.622167e+1	1.0000	0.2803
	Times optimized	30000	30000	39999.08	20000.92	40168.64	19832.36		
f_7	Mean	5.820803e-5	2.967570e+1	4.084439e-10	5.843126e+2	2.263991e-10	9.606294e+2		
	Median	2.677231e-5	4.335381e+0	2.411188e-10	3.426061e+2	1.064567e-10	2.8997e+2	0.0000	1.0000
	Times optimized	30000	30000	39999.24	20000.76	40614.24	19387		
f_8	Mean	3.189299e+5	1.462483e+2	2.04991e-12	1.090707e+3	4.783949e+5	1.258563e+3		
	Median	6.022649e-15	5.7762e+0	5.965865e-13	3.908341e+2	1.583456e-12	7.575727e+2	1.0000	1.0000
	Times optimized	30000	30000	39999.52	20000.48	40900.12	19100.84		

components. Conversely, the fitness value of the contribution based methods for the separable portion is higher than the corresponding value of their round-robin counterpart. Based on H_r we expect the test results for the non-separable portion to be close to zero and for the separable portion it is expected to be close to one. With a confidence interval of 90% ($\alpha = 0.1$) only the non-separable subcomponents of CBCC on f_5 and f_7 are significantly lower than DECC (p-value $< \alpha$). The reason for poor result of Wilcoxon test might be due to inherent inaccuracy of delta grouping in capturing interacting variables. Also note that in delta grouping there are 19 separable subcomponents, so the non-separable subcomponent is executed approximately 10 times less than ideal grouping. This might be another reason for poor results of Wilcoxon test. As a result of this, we expect to see a boost in the performance of CBCC when ideal grouping is employed.

Table 4 contains similar information to Table 3, however the ideal grouping is used. The overall trend in Table 4 is that the contribution based technique focuses most of the computational budget on non-separable subcomponents. For this reason the fitness of the non-separable portion is lower in CBCC-I implementations as opposed to DECC-I. This observation holds with a significance of 90% confidence interval. However, by looking at Table 2 we can see that this behavior does not necessarily result in a better overall performance. The reason can be identified by a closer look at the entries of f_6 and f_7 in Table 4. As it can be seen, the non-separable portions of the fitness function are relatively easy to optimize and the fitness value drops below the fitness value of the separable por-

tions. This behavior of the non-separable subcomponent causes its fitness to be negligible with respect to the separable portion. On the other hand, the round-robin implementation spends more time on the separable subcomponent which has the maximum effect on the global fitness value. This is why the overall fitness value of CBCC-I on f_6 and f_7 is lower than DECC-I. This behavior shows that CBCC is highly influenced by historical information in the early stages of evolution and it is not capable of responding to the local changes in fitness of the non-separable and separable subcomponents. In the case of f_6 and f_7, the non-separable subcomponent has the maximum *initial* contribution in the improvement of the global fitness. However, the early convergence of its value below the fitness value of the separable portion makes it negligible as compared to the separable portion of the fitness. This behavior can be clearly seen from the convergence plots of f_6 and f_7 in Figure 1(c) and 1(d) respectively. It is clear that the convergence graph of the non-separable portion crosses the convergence graph of the separable component relatively early in evolutionary process. In such situations it is expected from CBCC to switch to the subcomponent which has more contribution to the global fitness. This is clearly not the case for CBCC. With a better adaptation technique it is expected that contribution based method has the potential to outperform MA-SW-Chains on more functions. Overall, the convergence plots of f_6 and f_7 show that the non-separable portion's fitness value of CBCC is generally lower than DECC and conversely, the separable portion's fitness of CBCC is generally higher than

DECC. This is because of the emphasis that CBCC places on the subcomponent with the most contribution to the global fitness.

Another interesting observation is the occasional increases in the fitness of the separable component of the fitness. This can be clearly seen in Figures 2(a) and 2(e). It is notable that the global fitness is monotonically decreasing, however if we look at the fitness for the separable and non-separable portions separately we can see some occasional increases in their fitnesses. Interestingly, this behavior only happens when the delta grouping method is employed. It should be realized that when delta grouping is used, there are always a few non-separable variables mixed with the separable variables in different subcomponents. For example when a non-separable subcomponent undergoes mutation and crossover, since some of the separable variables are mixed with the non-separable ones, their value might also change in the process. These changes in the separable variables is not guaranteed to improve the fitness of the separable component, because the algorithm judges the fitness of an individual based on its global fitness and it does not distinguish between the fitness of the separable and non-separable portions. As a result of this, if there is a major improvement in the non-separable portion and yet there is a smaller increase in the fitness of the separable portion, the overall effect would be an improvement in the global fitness. The reason that we see this behavior is because we record the fitness of the separable and non-separable components separately.

As it was mentioned earlier, this behavior never happens in an ideal grouping situation. The reason is, when a subcomponent undergoes mutation and crossover – under ideal grouping – it only affects the separable *or* the non-separable portions at a time. This observation by itself is yet another motivation that encourages research in finding better decomposition strategies and moving towards automatic ideal grouping of the decision variables. Under such ideal decomposition the best individuals will be recognized and preserved and the computational budget will not be wasted by losing the best individuals in various subcomponents.

7. CONCLUSION

In this paper we have identified the issue of imbalance between the contribution of different subcomponents in cooperative co-evolution. Round-robin switching of subcomponents is only effective when there is no imbalance between the subcomponents. However, in many problems there may be an imbalance between the fitness contributions of different subcomponents. In such situations, using round-robin switching strategy results in wasting considerable amount of computational resources. To mitigate the imbalance issue we proposed Contribution Based Cooperative Co-evolution that – unlike ordinary cooperative co-evolution – selects the subcomponents based on their contribution in the improvement of the global fitness. By employing CBCC, less computational resources would be allocated to the subcomponents with negligible contributions.

One major drawback of CBCC is its slow response to local changes in the fitness value and relies on the information accumulated form the early stages of evolution. This weakness of CBCC could be resolved by using an adaptive technique that maintains a balance between the local and global contributions of subcomponents. In our future works, we are interested in investigating a proper adaptive strategy for CBCC in order to make it responsive to the local changes in the fitness contributions of various subcomponents.

Contribution Based Cooperative Co-evolution relies on a grouping strategy that captures the non-separable variables in a common subcomponents. In this paper we used delta grouping and ideal grouping for the analysis of CBCC. The promising results of an ideal grouping motivates further investigation in finding better grouping strategies for non-separable problems.

8. REFERENCES

[1] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. ser. Dover Books on Mathematics. Oxford University Press, 1996.

[2] R. E. Bellman. *Dynamic Programming*. ser. Dover Books on Mathematics. Princeton University Press, 1957.

[3] 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*, volume 6239 of *Lecture Notes in Computer Science*, pages 300–309. Springer Berlin / Heidelberg, 2011.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[5] W. S. Klug, M. R. Cummings, C. Spencer, C. A. Spencer, and M. A. Palladino. *Concepts of Genetics*. Pearson, 9 edition, 2008.

[6] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi. Scaling up fast evolutionary programming with cooperative coevolution. In *Proceedings of Congress on Evolutionary Computation*, pages 1101–1108, 2001.

[7] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1994.

[8] 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*, pages 3153–3160, july 2010.

[9] M. N. Omidvar, X. Li, and X. Yao. Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1762–1769, 2010.

[10] M. A. Potter and K. A. De Jong. A cooperative coevolutionary approach to function optimization. In *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, volume 2, pages 249–257, 1994.

[11] R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1995.

[12] 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. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2009. http://nical.ustc.edu.cn/cec10ss.php.

[13] F. van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation 8 (3)*, 2:225–239, 2004.

[14] Z. Yang, K. Tang, and X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178:2986–2999, August 2008.

[15] Z. Yang, K. Tang, and X. Yao. Self-adaptive differential evolution with neighborhood search. In *Proc. of IEEE Congress on Evolutionary Computation (CEC)*, pages 1110–1116, 2008.
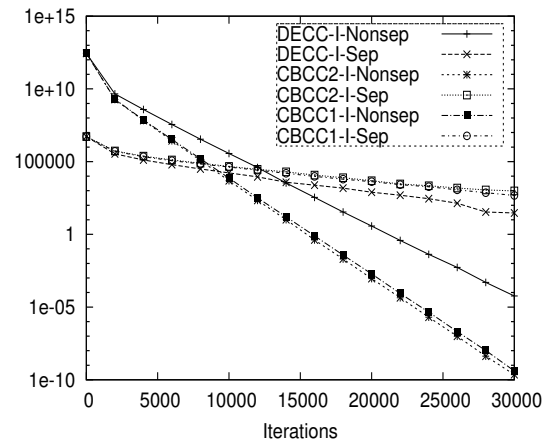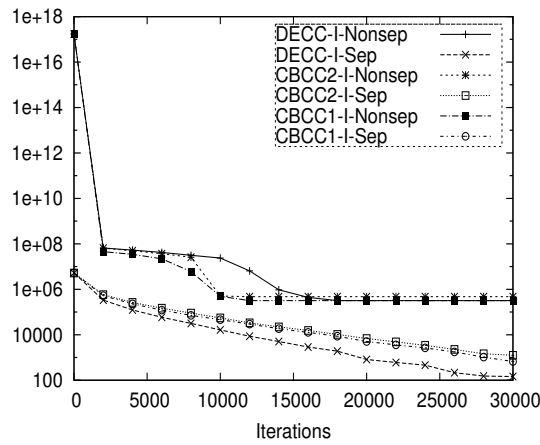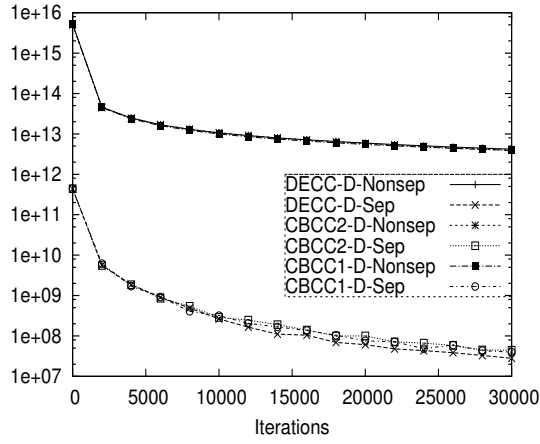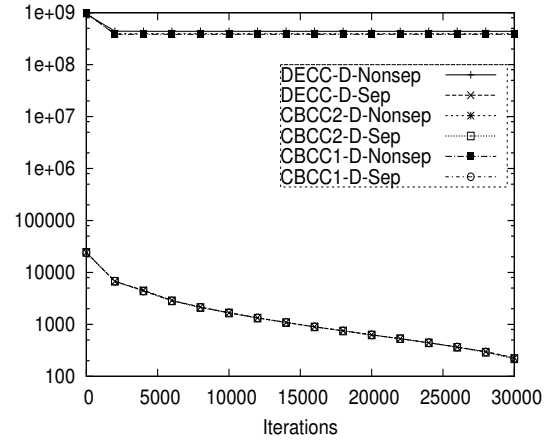
(a) f_4

(b) f_5
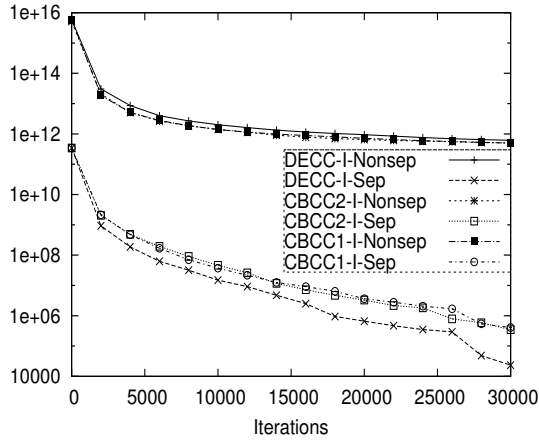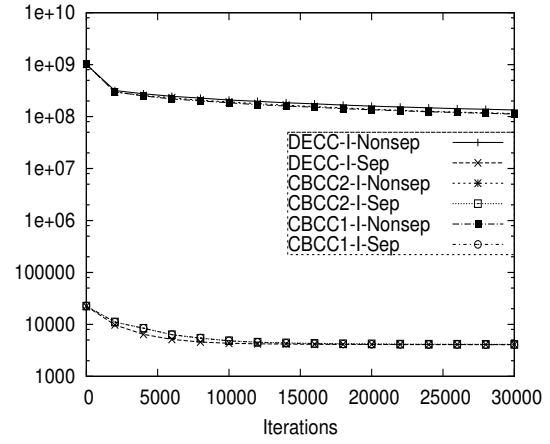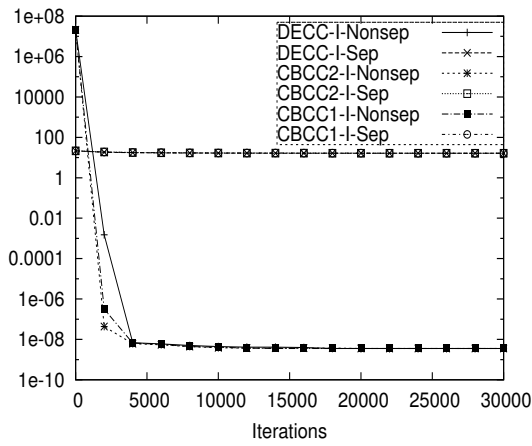
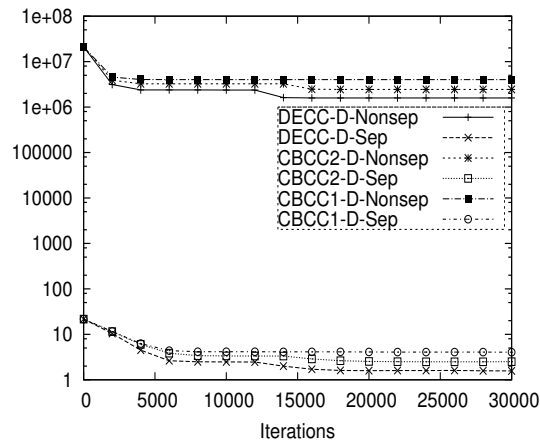(c) f_6

(d) f_7

(e) f_8

Figure 1: Convergence plots of f_4 to f_8 using ideal grouping. The plots are based on average of 25 independent runs.
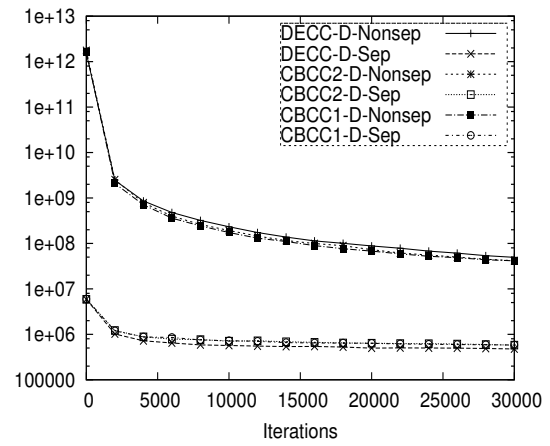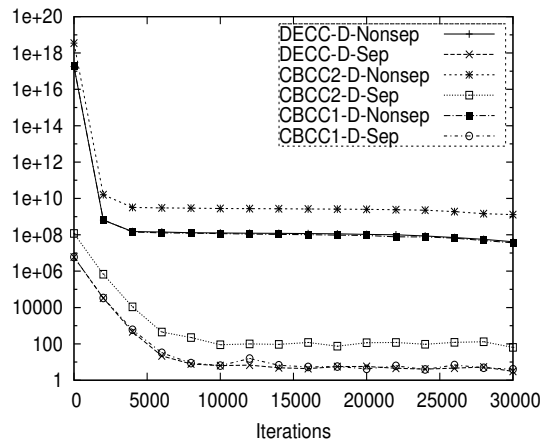
(a) f_4

(b) f_5

(c) f_6

(d) f_7

(e) f_8

Figure 2: Convergence plots of f_4 to f_8 using delta grouping. The plots are based on average of 25 independent runs.