Impact of Problem Decomposition on Cooperative Coevolution

Wenxiang Chen* and Ke Tang†

*Computer Science Department, Colorado State University
1873 Campus Delivery, Fort Collins, CO 80523, United States
Email: chenwx@cs.colostate.edu

† USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications
School of Computer Science and Technology, University of Science and Technology of China
Hefei, Anhui, China, 230027
Email: ketang@ustc.edu.cn

Abstract—Variable Interaction Learning (VIL) is an emerging technique regarding detecting interacting variables so that Cooperative Coevolutionary Evolutionary Algorithms (CCEAs) can decompose problems accordingly and tackle subproblems of smaller sizes. While previous approaches are developed to efficiently perform VIL, no study has been on the actual usefulness of the detected variable interactions in terms of the performance of CCEAs. Since VIL is a computationally expensive task by itself, overly spending time on VIL without notable benefits for CCEAs should be avoided. It is hence critical to study the real impact of problem decomposition on CCEAs. We conduct empirical studies to address three closely related questions: 1) will a better problem decomposition lead to better performance of CCEAs, 2) when will improving problem decomposition benefit CCEAs, and 3) to what extent will improving problem decomposition enhance the performance of CCEAs.

I. INTRODUCTION

Cooperative Coevolutionary Evolutionary Algorithms (CCEAs) are invented to handle increasingly complex problem by evolving interacting co-adapted subproblems [1]. CCEAs employ a divide-and-conquer strategy, decompose the original problems into several subproblems and then optimize each subproblem separately. Many studies [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] state the ideal problem decomposition should group the mutually tightly interacting variables into the same subproblem, while keeping the interactions weak among distinct subcomponents.

However, the interacting variables is usually unknown beforehand. Variable Interaction Learning (VIL) is a newly developed technique on this issue [2]. VIL methods incrementally detect interactions between pairs of variables. The detected interactions during VIL are used to form a partition of decision variables. CCEAs can then decompose the problem accordingly. Previous studies invest great efforts on getting to the “optimal” problem decomposition as close as possible. Nevertheless, learning variable interactions consumes computation time and can be computationally expensive. The goal of evolutionary optimization is to obtain a solution of good quality in the given amount of time, rather than to successfully detect all variable interactions. In this paper, we argue that one should consider VIL from a cost-effectiveness perspective. To shed some lights along this line, we are going to empirically investigate the actual benefit of detecting variable interactions. By incorporating prior knowledge on variable interactions varying from none to complete, the actual benefit of detecting variable interactions on the performance of CCEA is uncovered in the current paper.

Our empirical studies will be directed by three research questions. The answers to these questions will provide valuable guidance for designing VIL methods. Previous studies regarding VIL focus on how to detect more variable interactions in shorter time. Yet the question really is, will more detected variable interactions necessarily lead to better performance of CCEAs? If not, one can simply terminate VIL after a certain stage, since to detect more variable interactions is then nothing but a waste of computational time. The questions that follows is, when will detecting more variable interactions help CCEAs? The answer to this yields when to stop VIL. Once knowing more variable interactions stops helping CCEAs, there is no need to invest efforts in detecting more variable interactions. In addition, we need to be more precise about to what extent will detecting more variable interactions benefit CCEAs. There can be situations where detecting any more variable interactions is very costly while the benefit of doing so is marginal. In these cases, one may choose to terminate VIL, which leaves more time for CCEAs.

II. BACKGROUND

A. Cooperative Coevolution

Cooperative Coevolutionary Evolutionary Algorithms (CCEAs) usually consist of three basic ingredients [8]:

1) A decomposition method used to divide the N-dimensional decision vector into m groups $G_1 \ldots G_m$ of variables. Each such group is optimized with a separate subpopulation of the corresponding dimension $|G_i| < N$. This also yields that the number of subpopulations equals the number of groups $m$.

2) In order to evaluate the fitness of the individuals from a certain subpopulation, a representative element from each of the other subpopulations is selected. In this cooperation step, a population of complete N-dimensional candidate solutions is constructed by concatenating the representatives to each element of the current subpopulation.
Problem decomposition can be viewed as a projection of joint space onto several subspace with lower dimensions. The way of projecting joint space is known to be related to the performance of CCEAs [13], [6], [7], [14], [15], [16]. CCEAs suffer from relative over-generalization [17], if the decomposition separates interacting variables into different subproblems, the projection throws considerable amount of information away. CCEAs converge to a suboptimal solution quickly in that case.

B. Variable Interaction Learning

To resolve the foregoing issue, Cooperative Coevolution with Variable Interaction Learning (CCVIL) [2] introduces a Variable Interaction Learning (VIL) stage prior to the execution of CCEAs. CCVIL is therefore a two-stage approach that first proactively detects variable interactions to form a partition of variables and then feeds the partition into a CCEA for problem decomposition. In the first stage (i.e., the VIL stage), each variable is initialized as an individual group, and then CCVIL samples the search space to collect evidence of variable interaction. Once any pair of variables from different groups are detected to be interacting, the two groups which the variables belong to are merged into a larger one. In other words, any detected variable interaction reduces the number of groups by exactly one.

The details about VIL are beyond the scope of this paper, and can be found at paper [2]. Instead of echoing the detailed procedure of CCVIL, we demonstrate the VIL stage of CCVIL in figure 1 on a small problem with 6 variables. VIL initializes each variable as an individual group, which lead to 6 groups as the initial problem decomposition. VIL then checks every pair of consecutive variables in a random permutation for interaction. After checking the last pair of variables, another random permutation is generated and a new cycle starts. In this example, \( x_1 \) and \( x_2 \) is first found to be interacting, so the two groups \{1\} and \{2\} where \( x_1 \) and \( x_2 \) belong to are merged into a larger group \{1, 2\}. After performing two more merge operations, VIL finally reaches the target problem decomposition \{\{1, 2, 3\}, \{4, 5\}, \{6\}\}, which is passed to the second stage so that CCEAs can decompose the problem accordingly. Note that during the second stage of CCVIL, the problem decomposition is fixed.

III. MOTIVATIONS

Since VIL is about detecting variable interactions, better efficiency of VIL leads to discover more intrinsic (if not all) variable interactions. Nonetheless, current paper is not about pushing forward the efficiency of VIL. Instead, we attempt to answer a more fundamental question: will detecting more variable interactions necessarily result in a better performance of CCEAs? This is our first question. An answer to the question is required to lay the foundation for further exploring better VIL methods.

There are two opposing factors in applying CCEAs with VIL. First, the decomposing problem makes the subproblem more tractable to handle. Although any undetected variable interaction can lead to excessively divide the problem, since the two groups are supposed to merge. In this case, the problem decomposition results in the over-simplification of the original problem, which in return traps CCEAs into local optima. Second, merging two groups of variables to form a larger group makes the new subproblem harder to solve, because the size of the new subproblem is the sum of that of the previous two subproblems. It takes longer for CCEAs to converge in this scenario. It is possible that CCEA with better problem decomposition (lesser amount yet larger sizes of groups) reaches cutoff point in running time before converge and ends up with an inferior solution, compared with the one with worse problem decomposition which yet converges faster. As a result, we hypothesize that detecting more intrinsic variable interactions will not always enhance the performance of CCEAs. Based on this, our second question is, when can detecting more variable interactions benefit CCEAs?

While VIL is able to claim the existence of variable interactions once it finds the evidence, it cannot judge the non-existence of variable interactions. A pair of variables that have not been found interacting in VIL does not necessarily suggest that they are independent from each other. It is entirely possible that the evidence of interaction is there, yet have
not been discovered by then. Also, even state-of-the-art VIL methods often fail to discover all intrinsic variable interactions in an acceptable amount of time [2]. Thereby, the variable interactions detected by VIL are likely to be a portion of all the intrinsic variable interactions. Every time one more variable interaction is detected, the partition suggested by VIL becomes closer to the target partition. Although there is always a cost (which of course relates to the specific VIL method in use) associated with the detection of each variable interaction. One might wonder, is it really worthwhile to pay the cost for detecting more variable interactions? The answer relies on the specific VIL method: what exactly is the benefit (if any) of detecting more variable interactions? The answer depends on the ideal problem decomposition that have not been proved interacting as independent ones. Increasing $P_{prior}$ perfectly simulates the circumstance where more variable interactions (if not all) are detected as time elapses.

V. Empirical Studies

A. Experimental Setup

- **Benchmark set**: Four $\frac{D}{m}$-group $m$-nonseparable functions from CEC’2010 Special Session [21]. The dimensions of the problems are all $D = 1000$, and the dimension of each non-separable problem is $m = 50$. In other words, the ideal problem decomposition will contain exactly $\frac{D}{m} = 20$ groups. Details about the benchmark set can be found later in the subsequent subsection.

- **Optimizer**: CCVIL from paper [2] without the VIL stage, the decomposition is generated using prior knowledge from $A_{intr}$.

- **Maximum number of fitness evaluation for each run**: 3 million.

- Results were collected based on 25 independent runs on each function.

B. Benchmark Set

In this subsection, we review the four functions selected from the benchmark suite provided by the CEC’2010 Special Session on Large Scale Global Optimization [21]. We provide some notations followed by the definition of the selected benchmark functions.

**Notations:**

- **Dimension**: $D = 1000$
- **Group size**: $m = 50$
- **$x = (x_1, x_2, \cdots, x_D)$**: the candidate solution – a $D$-dimensional row vector
- **$o = (o_1, o_2, \cdots, o_D)$**: the (shifted) global optimum
- **$z = x − o$, $z = (z_1, z_2, \cdots, z_D)$**: the shifted candidate solution – a $D$-dimensional row vector
- **$P$: random permutation of $\{1, 2, \cdots, D\}$

**Function definitions:**

- **$\frac{D}{m}$-group shifted and $m$-rotated Elliptic Function**

  \[
  f_{14}(x) = \sum_{k=1}^{\frac{D}{m}} f_{rot\_elliptic}[z(P_{(k-1)m+1} : P_{km})]
  \]

- **$\frac{D}{m}$-group shifted and $m$-rotated Rastrigin’s Function**

  \[
  f_{15}(x) = \sum_{k=1}^{\frac{D}{m}} f_{rot\_rastrigin}[z(P_{(k-1)m+1} : P_{km})]
  \]

- **$\frac{D}{m}$-group shifted and $m$-rotated Ackley’s Function**

  \[
  f_{16}(x) = \sum_{k=1}^{\frac{D}{m}} f_{rot\_ackley}[z(P_{(k-1)m+1} : P_{km})]
  \]
• \( \frac{D}{m} \) group shifted \( m \)-dimensional Schwefel’s Problem 1.2

\[
f_{17}(x) = \sum_{k=1}^{D} f_{\text{schwefel}}\left[ z(P_{(k-1)m+1} : P_{km}) \right]
\]

These four benchmark functions are selected because the intrinsic structures of them are all partially-separable, in which CCEAs with correct problem decomposition excel. They provide a platform for studying how partially correct problem decomposition can affect the performance of CCEAs. Also note that the evaluation values for global optima for the four benchmark functions are all zeros, and we are minimizing the functions.

**C. Empirical Results with \( P_{\text{prior}} \) varying from 0% to 100% with step size 10%**

Intuitively, we set the range of \( P_{\text{prior}} \) to be \([0\%, 100\%]\), since the dynamics of CCEA’s performance when prior knowledge ranges from none to complete are supposed to be uncovered. Limited by computational resources, we divide the interval \([0\%, 100\%]\) by 10%. Comprehensive experimental results are presented in table I.

Table I implies: 1) Incorporating prior grouping knowledge generally enhances the performance of CCEAs. However, raising \( P_{\text{prior}} \) appears to be useful mostly within the interval \([0\%, 10\%]\). The underlying reason for this counter-intuitive observation will be uncovered in Subsection V-E. 2) There are indeed cases where increasing \( P_{\text{prior}} \) impairs CCEAs, even though the difference is marginal. For instance, the mean value of solutions found by CCEA with \( P_{\text{prior}} = 50\% \) is \(5.54e+06\), which is even lower than that with \( P_{\text{prior}} = 100\%\), \(5.78e+06\). 3) The best solution while varying \( P_{\text{prior}} \) is rarely with \( P_{\text{prior}} = 100\% \). This suggest the optimal problem decomposition may not necessarily yield the best solution. In most cases, partially correct ones can instead explore slightly better potential of CCEAs. Therefore, the answer to our first question is: detecting more variable interactions can help CCEAs find better solutions, but it is not always true. 4) Our hypothesis regarding worse problem decomposition can sometimes yield solutions of better quality is also verified.

Furthermore, the results clearly suggest that the sampling points for \( P_{\text{intr}} \) should be shrunk on the interval of \([0\%, 10\%]\), in order to provide finer insight into the dynamics of CCEA’s performance while varying \( P_{\text{intr}} \).

**D. Empirical Results with \( P_{\text{prior}} \) varying from 1% to 9% with step size 1%**

In this subsection, the step size of sampling point is reduced to be 1%, and the range of interest is restricted in \([1\%, 9\%]\). The empirical results with detailed statistics are shown in table II.

We observe from table II that: 1) with \( P_{\text{prior}} \) changing from 1% to 9%, the performance of CCEAs increase mostly monotonically. It suggests that when the acquired knowledge regarding problem structure is sparse (i.e., from 1% to 9% in our case), detecting more variable interactions almost always enhance CCEAs. 2) The improvement is as much as several orders of magnitude in fitness can be achieved by only using 10% of prior knowledge. It is indeed worthwhile to invest efforts into learning variable interactions, at least to a certain extent.

**E. Insights into Relationship between Detecting Variable Interactions and Problem Decomposition**

This subsection is going to provide some more explanations into the dynamics of the performance of CCEA by associating \( P_{\text{intr}} \), fitness of obtained solutions and problem decomposition strategies all together in table III.

The combined results in table III show that 1) Given \( P_{\text{intr}} = 9\% \), the corresponding decomposition strategies, which yield around 25 subpopulations, is already quite close to the optimal one that is supposed to have 20 subpopulations. It is the reason why further increase of \( P_{\text{intr}} \) make little progress in enhancing the performance of CCEA, as shown previously in table I. 2) The relationship between \( P_{\text{intr}} \) and the number of subpopulations is nonlinear. More precisely, the speed of subpopulations being merged decrease greatly with the \( P_{\text{intr}} \). Figure 2 illustrates the relationship. The trend is consistent among all four benchmark functions. The empirical data perfectly matches our expectation: merging subpopulations gradually reduces the number of undiscovered interactions, which in return, makes it harder to further merge more subpopulations.

In order to better understand the impact of problem decomposition on CCEAs, we also plot the convergence curve for each benchmark function in figure 3. It shows that: 1) When the prior knowledge regarding the intrinsic variable interactions is sparse (i.e., \( P_{\text{prior}} \) from 1% to 9%), the convergence curves of the CCEAs associated with higher \( P_{\text{prior}} \) appears to dominate the ones with lower \( P_{\text{prior}} \) (since there is nearly no crossover point among convergence curves). 2) Our previous hypothesis regarding CCEAs with worse problem decomposition can end up with better suboptimal does not hold when \( P_{\text{prior}} \) is small (from 0% to 10%).

**VI. Conclusion and Future Work**

We empirically investigate the impact of problem decomposition on the performance of CCEAs in this paper. We first hypothesize that detecting more variable interactions does not necessarily result in performance of CCEAs. Our study is then motivated by three related questions: 1) will detecting more variable interactions lead to better performance of CCEAs? If not, 2) when will detecting more variable interactions benefit CCEAs? 3) what exactly is the benefit of detecting more variable interactions for CCEAs? From the empirical data, the following conclusions can be drawn: 1) Detecting more variable interactions are generally beneficial to CCEAs and yet may also deteriorate the performance of CCEAs. 2) When the acquired knowledge about intrinsic variable interactions is meager (less than 10% of the total interactions), knowing more intrinsic variable interactions does monotonically improve the performance of CCEAs. 3) By detecting only 10% of knowledge on variable interactions, the fitness of solutions obtained by CCEAs can be improved by several orders of magnitude. Our study lays the necessary foundation for variable interaction learning and also provides valuable guidance for developing new variable interaction learning methods.
TABLE I. EXPERIMENTAL RESULT WITH $P_{\text{prior}}$ VARYING FROM 0% TO 100% WITH STEP SIZE 10%. NUMBERS IN THE FIRST LINE STAND FOR THE PERCENTAGE OF THE GIVEN INTERACTION INFORMATION. THE RESULTS FOR THE BEST, 7th, median, 14th, worst runs as well as the mean and the standard deviation over 25 independent runs are presented. Best solutions among all settings of $P_{\text{prior}}$ are marked in **BOLD**.

<table>
<thead>
<tr>
<th>$P_{\text{prior}}$</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>4.74e+08</td>
<td>5.99e+08</td>
<td>4.78e+08</td>
<td>4.75e+08</td>
<td>5.38e+08</td>
<td>5.93e+08</td>
<td>5.92e+08</td>
<td>5.74e+08</td>
<td>5.58e+08</td>
<td>5.24e+08</td>
<td>4.07e+08</td>
</tr>
<tr>
<td>7th</td>
<td>5.64e+08</td>
<td>6.99e+08</td>
<td>5.42e+08</td>
<td>5.38e+08</td>
<td>6.32e+08</td>
<td>5.17e+08</td>
<td>5.46e+08</td>
<td>5.27e+08</td>
<td>5.62e+08</td>
<td>5.34e+08</td>
<td></td>
</tr>
<tr>
<td>med.</td>
<td>5.97e+08</td>
<td>7.63e+08</td>
<td>6.18e+08</td>
<td>5.80e+08</td>
<td>6.67e+08</td>
<td>5.66e+08</td>
<td>6.49e+08</td>
<td>5.97e+08</td>
<td>5.59e+08</td>
<td>6.32e+08</td>
<td>5.64e+08</td>
</tr>
<tr>
<td>19th</td>
<td>6.29e+08</td>
<td>8.14e+08</td>
<td>6.42e+08</td>
<td>6.23e+08</td>
<td>7.10e+08</td>
<td>5.98e+08</td>
<td>6.97e+08</td>
<td>6.37e+08</td>
<td>6.45e+08</td>
<td>6.75e+08</td>
<td>6.23e+08</td>
</tr>
<tr>
<td>max</td>
<td>7.32e+08</td>
<td>1.09e+09</td>
<td>7.73e+08</td>
<td>7.35e+08</td>
<td>7.94e+08</td>
<td>6.37e+08</td>
<td>7.48e+08</td>
<td>7.63e+08</td>
<td>7.95e+08</td>
<td>9.13e+08</td>
<td>7.27e+08</td>
</tr>
<tr>
<td>mean</td>
<td>5.97e+08</td>
<td>7.64e+08</td>
<td>6.03e+08</td>
<td>5.81e+08</td>
<td>6.19e+08</td>
<td>5.96e+08</td>
<td>6.39e+08</td>
<td>5.92e+08</td>
<td>6.15e+08</td>
<td>5.77e+08</td>
<td>5.75e+08</td>
</tr>
<tr>
<td>std</td>
<td>5.98e+07</td>
<td>1.17e+08</td>
<td>6.35e+03</td>
<td>7.30e+05</td>
<td>6.88e+05</td>
<td>5.79e+05</td>
<td>6.43e+05</td>
<td>5.79e+05</td>
<td>9.15e+05</td>
<td>1.07e+06</td>
<td>7.16e+05</td>
</tr>
</tbody>
</table>

**TABLE II. EXPERIMENTAL RESULTS WITH $P_{\text{prior}}$ VARYING FROM 1% TO 9%. NUMBERS IN THE FIRST LINE STAND FOR THE PERCENTAGE OF THE GIVEN INTERACTION INFORMATION. THE RESULTS FOR THE BEST, 7th, median, 14th, worst runs as well as the mean and the standard deviation over 25 independent runs are presented.**
Current work is also limited in a couple of ways. First, the variable interactions under our consideration is pairwise ones. We merge two groups into a larger one that respectively include interacting variables by assuming pairwise interactions exists among every pair of variables in the new group. This may not always be the case. An exception is the generalized Rosenbrock function [22], in which the variable interactions form a chain and two non-consecutive variable in the chain do not have any pairwise interaction. Second, our empirical study is limited on a special type of structured problems, in which the sizes of groups are all equal. There is still a doubt about whether our conclusion can be generalized well to problems of other types of structures.

**ACKNOWLEDGMENT**

This research was partially sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number FA9550-11-1-0088, the National Natural Science Foundation of China (Grant No. 61175065) and the Natural Science Foundation of Anhui Province (Grant No. 1108085J16). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.
Fig. 3. Convergences curves for each benchmark function when $P_{\text{prior}}$ is in [1%, 9%].

REFERENCES


