Cooperative Co-evolution by Decomposition Optimization based on LINC-R with EGA for Large scale Optimization

Rui Zhong[1] and Masaharu Munetomo[2]

[1] Graduate School of Information Science and Technology, Hokkaido University,
Sapporo, 060-0814, Japan
[2] Information Initiative Center, Hokkaido University, Sapporo, 060-811, Japan
rui.zhong.u5@elms.hokudai.ac.jp
munetomo@iic.hokudai.ac.jp

Abstract. Linkage Identification by Nonlinearity Check for Real-Coded Genetic Algorithms (LINC-R) and Differential Grouping (DG) as a basic strategy for large-scale problem decomposition has been widely extended in various techniques. As a popular decomposition strategy, the LINC-R/DG-based decomposition strategy has the advantages of full automation and high accuracy and has achieved exciting results on multiple test suites. However, the disadvantages are obvious: (1) An amount of fitness evaluation times (FEs) is required to identify interactions between variables. (2) Local linearity/non-linearity is detected to represent global linearity/non-linearity, which makes LINC-R/DG-based decomposition methods limited. In this paper, we propose a novel decomposition algorithm that treats the decomposition problem as an optimization problem and designs the objective function based on LINC-R/DG. We mathematically explain the feasibility of our objective function. Experimental studies show that our decomposition strategy has broad prospects for solving large-scale optimization problems and can be applied to solving higher-dimensional problems by simple expansion.

Keywords: Cooperative Co-evolution (CC), Linkage Identification, Large-scale Problem, Decomposition Optimization

1 Introduction

Large-scale optimization problems widely exist in data science[18] and engineering[5, 2]. At present, research on large-scale optimization problems mainly includes two goals, solving optimization problems with higher dimensions and proposing better algorithms for current high-dimensional problems. Meanwhile, algorithms that perform well on low-dimensional problems often approach poor performance on high dimensions. This is mainly due to the following reasons: (1) The complexity of optimization problems tends to increase with the dimension increases. (2) The search space of large-scale problems increases exponentially with the increase of dimension, known as the curse of dimensionality[6]. (3) The computational cost of building the surrogate model is expensive, and the accuracy also decreases which makes some algorithms (Estimation of Distribution Algorithms[1, 17] (EDAs), Bayesian Optimization Algorithm[16] (BOA), etc.) limited in large-scale problems.

The cooperative co-evolution[19] (CC) framework was first proposed to solve large-scale optimization problems based on divide and conquer. The basic idea of CC is to decompose the original problem into multiple non-separable subproblems, and then apply Evolutionary Algorithms (EAs) to solve sub-problems separately, which has achieved great success in solving large-scale continuous[12], combinatorial[9], constrained[20], and multi-objective[23]. However, several studies[21, 14, 15] have shown that the CC framework is sensitive to the problem decomposition and how to design the decomposition methods become a popular research topic.

In this paper, we propose a novel decomposition method named Decomposition Optimization based on LINC-R with EGA(EGALINC-Rmin), our proposal method allows an automatic decomposition method that treats the decomposition problem as an optimization problem and apply the Elitist Reservation GA (EGA) to optimize the objective function. We test our proposal in CEC2013 Large-scale Global Optimization (LSGO) suite[7] and comparing with DECC-D[13], DECC-G[25] and DECC-DG[12] to illustrate the superiority of our proposal in solving high-dimensional problems.

The rest of this paper is organized as follows. Section II describes the gene interaction and introduces the CC framework with a brief review of popular decomposition methods. Section III describes the proposal in detail. Section IV describes the experiments and analyzes the experimental results. Section V concludes the paper and shows future directions.

2 Related works

This section introduces the concept of Gene interaction and how the CC framework works and briefly reviews the currently popular decomposition methods.

2.1 Gene interaction

The correlation between genes is known as Epistasis[10] or gene interaction[12]. In biology, if a feature at the phenotype level is determined by two or more genes, then we call these genes have interaction, and the genome composed of these genes is called the linkage set[24]. We extend the concept of gene interaction to the definition of optimization problems. When $\min(f(x_1, x_2, ..., x_n))$ =min $(\sum_{i=1}^{m} f(x_{i_1}, ..., x_{i_k})), L_i = \{i_1, ..., i_k\}$ is defined as linkage set, and the same varible is possible in different linkge set, which is called overlap variable. In this situation, we call $f(x)$ is a partially separable function. The interaction in the linkage set exists in every pair of variables, and we call these variables non-separable variables. There is no interaction between variables in different linkage sets (except for overlapping variables), and these variables are called separable variables. There are two extreme cases of the interactions between variables. When there is no interaction between all variables, $\min(f(x_1, x_2, ..., x_n))$ $=\min(\sum_{i=1}^{m} f(x_{i_i}))$, we call $f(x)$ is a fully separable function. On the contrary, when every pair of variables exist the interaction, we call $f(x)$ a fully non-separable function. Note that there are two cases of fully non-separable functions, one is a fully non-separable function with direct interactions between every pair of variables, and the other is existing indirect interactions. Taking a problem with 5 variables as an example, the schematic diagram is shown in Figure 1.

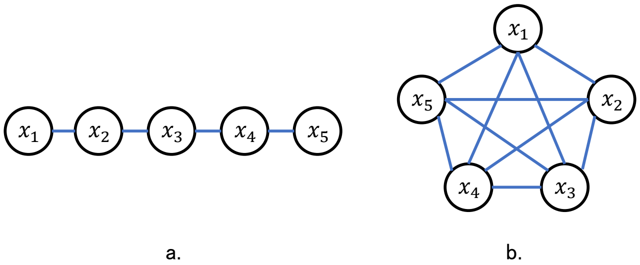

Fig. 1. Different structures of fully non-separable functions contain 5 variables. a). a fully non-separable function with indirect interactions. b). a fully non-separable function without indirect interactions

In Figure 1(a), although there is no direct interaction between x_1 and x_3, since x_1 and x_2 have interaction, x_2 and x_3 have interaction, we think that there is an indirect interaction between x_1 and x_3. In Figure 1(b), any variable interacts with the rest of the variables, so it is easy to identify the problem in Figure 1(b) as non-separable function.

2.2 CC framework and brief review

Based on the divide and conquer, CC framework decompose the large-scale optimization problem into multiple non-separable sub-problems and optimize separately. A perfect decomposition can reduce the search space expoentially without losing accuracy in optimization, therefore, CC framework is one of the most popular strategies for solving large-scale optimization problems. Cooperative Coevolutionary Genetic Algorithm[19] (CCGA) was proposed with the CC framework. In CCGA, the simplest decomposition method is adopted to decompose n-D problems into $n \times 1$-D sub-problems. This decomposition method completely ignores the interactions between variables. In theory, this strategy will be far better than standard EAs on fully separable functions, but it will be completely unacceptable for fully non-separable functions. The experimental results verify this assumption. Random grouping[25] is proposed without fitness landscape information. This method randomly decomposes an n-D problem into $m \times k$-D subproblems ($n = m \times k$), where m and k are decided by the user. In Large Scale Evolutionary Optimization Using Cooperative Coevolution (DECC-G), it shows mathematically that the probability of two interaction variables in the same sub-problem is fairly high across multiple trial experiments in random decomposition. Experiments show that DECC-G performs well on problems up to 1000-D.

Efficient decomposition is impossible with little fitness landscape information. The proposal of Delta grouping[13] (DECC-D) alleviates this problem. DECC-D realizes that the difference in coordinates from the initial random population to the optimized population is larger when x_i and x_j are separable variables. Based on this consideration, DECC-D optimizes the random population for one generation, calculates the difference of each dimension, and decomposes the variable into $m \times k$-D sub-problems ($n = m \times k$) according to the sorted dimension difference. The experimental results on CEC2008 and CEC2010 show that this rough estimation method can better capture interaction variables with little prior knowledge.

Linkage Identification by Non-monotonicity Detection for Overlapping Functions[11] (LIMD) and Linkage Identification by Nonlinearity Check for Real-Coded Genetic Algorithms[24] (LINC-R) are proposed for higher decomposition accuracy. When LIMD detects the interaction between x_i and x_j, the following strategy is applied

$$\begin{aligned} s &= (x_1, x_2, ..., x_n) \\ s_i &= (x_1, ..., x_i + \delta, ..., x_n) \\ s_j &= (x_1, ..., x_j + \delta, ..., x_n) \\ s_{ij} &= (x_1, ..., x_i + \delta, ..., x_j + \delta, ..., x_n) \end{aligned} \tag{1}$$

Eq (2) shows the mechanism of LIMD

$$\begin{aligned} if\ \forall s \in Pop:\ & f(s) < f(s_i) < f(s_{ij})\ and\ f(s) < f(s_j) < f(s_{ij}) \\ or\ & f(s) > f(s_i) > f(s_{ij})\ and\ f(s) > f(s_j) > f(s_{ij}) \\ & then\ x_i\ and\ x_j\ are\ separable \end{aligned} \tag{2}$$

Pop denotes random population with proper size. LIMD detects the monotonicity of the function at multiple samples. When x_i and x_j satisfy the simultaneous increase or decrease at all samples, the LIMD identifies x_i and x_j as separable variables. Cooperative coevolution with variable interaction learning[3] (CCVIL) also investigates the monotonicity of variables at samples, proposing similar rules:

$$\begin{aligned} if \exists s, s_i, s_j, s_{ij}, :\ & f(s) > f(s_i)\ and\ f(s_j) < f(s_{ij}) \\ & then\ x_i\ and\ x_j\ are\ non-separable \end{aligned} \tag{3}$$

Based on eq (3), CCVIL proposes perturbations to multiple dimensions at the same time, which allows the algorithm to detect the interaction between a variable with the entire sub-problem, and has shown good performance on the CEC2010 LSGO benchmark suite.

LINC-R slightly perturbs in x_i and x_j and identifies by calculating the difference in fitness. Particularly,

$$\begin{aligned} \Delta f_i &= f(s_i) - f(s) \\ \Delta f_j &= f(s_j) - f(s) \\ \Delta f_{ij} &= f(s_{ij}) - f(s) \\ if|\Delta f_{ij} - (\Delta f_i + \Delta f_j)| &< \epsilon \\ then\ x_i\ and\ x_j\ are\ separable \end{aligned} \tag{4}$$

LINC-R applies eq(4) to the low-dimensional test functions with a random population and receives the inspired results. Differential grouping (DG) is proposed to detect the interaction between variables based on the LINC-R. In the DG paper, eq(4) is applied to the problems up to 1000 dimensions and shows the exciting experimental results on the CEC2010 suite. At present, the LINC-R/DG-based decomposition methods are considered as high-accuracy decomposition methods. graphDG[8] even achieved 100% accuracy on the most of test functions in the CEC2010 LSGO suite. However, the disadvantage of high accuracy decomposition is that it requires an amount of computational cost, although several current studies have reduced the computational cost from $O(n^2)$[12] to $O(nlog(n))$[22] or even lower[14], It is unacceptable in higher-dimensional problems (such as millions, tens of millions). And due to the expensive computational cost, the LINC-R/DG-based decomposition method often only represents the global interaction by the local interaction check around a certain sample, which makes these methods limited.

3 EGALINC-Rmin

In this section, we will introduce the details of our proposal and the techniques. The flowchart of our proposal is shown in Figure 2.

How to evaluate the decomposition solution is the main problem of our research. In the past research, a decomposition method named Fitness Difference Minimization[21] was proposed. First, the n-D problem was decomposed into $m \times k$-D problems ($n = m \times k$) by the user, and the decomposition problem was regarded as an optimization problem. According to eq (5)

$$\min(f(x) - \sum_{i=1}^{m}(f_i(x_i))^2) \tag{5}$$

which $f_i(x_i)$ is the sub-problem which contains k variables.

In our previous work[26], we design the objective function based on LINC-R, and the details are as follows.

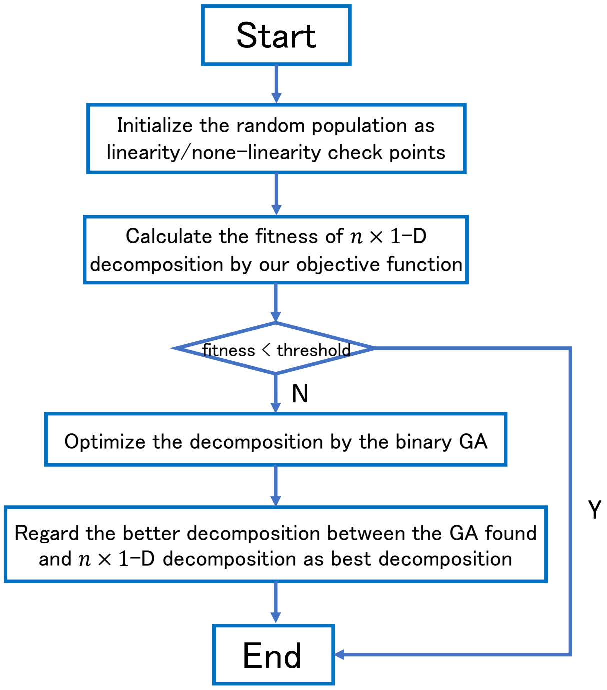
Start
Initialize the random population as linearity/none−linearity check points
Calculate the fitness of $n \times 1$−D decomposition by our objective function
fitness < threshold
N
Y
Optimize the decomposition by the binary GA
Regard the better decomposition between the GA found and $n \times 1$−D decomposition as best decomposition
End

Fig. 2. The flowchart of our proposal

In LINC-R, we can rewrite eq (4) to eq (6)

$$\begin{aligned} if|(f(s_{ij}) - f(s_i)) - (f(s_j) - f(s))| < \epsilon \\ then\ x_i\ and\ x_j\ are\ separable \end{aligned} \tag{6}$$

Figure 3 shows the mechanism of LINC-R works on both separable and non-separable variables as our previous work[26].

In our proposal, a variant of LINC-R is applied. Eq 6 can be written as

$$\begin{aligned} if|(f(s_{ij}) - f(s)) - ((f(s_i) - f(s)) + (f(s_j) - f(s)))| < \epsilon \\ then\ x_i\ and\ x_j\ are\ separable \end{aligned} \tag{7}$$

According to Figure 4, we technically transform LINC-R to the additive form of vectors [26]. When eq 7 is satisfied, x_i and x_j are separable variables

Based on this interesting finding, we next derive LINC-R to 3-D or higher dimensions. In 3-D space, the schematic diagram is shown in Figure 5.

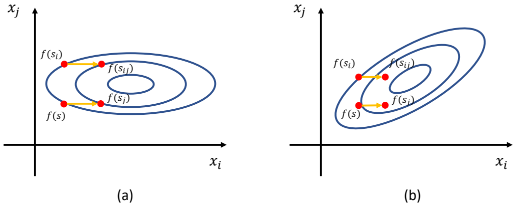

Fig. 3. (a).LINC-R works on the separable variables. (b).LINC-R works on the non-separable variables.

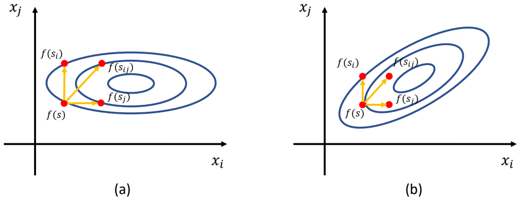

Fig. 4. (a) The transformation LINC-R works on the separable variables. (b) The transformation LINC-R works on the non-separable variables.

Similarly[26], if eq (8) is satisfied, then x_i, x_j, and x_k are fully separable.

$$\begin{aligned} \Delta f_i &= f(s_i) - f(s) \\ \Delta f_j &= f(s_j) - f(s) \\ \Delta f_k &= f(s_k) - f(s) \\ \Delta f_{ijk} &= f(s_{ijk}) - f(s) \\ if|\Delta f_{ijk} - (\Delta f_i + \Delta f_j + \Delta f_k)| &< \epsilon \\ then\ x_i, x_j, x_k\ are\ fully\ separable& \end{aligned} \tag{8}$$

Therefore, we can reasonably infer that when the dimension reaches n, if eq (9) is satisfied, then $f(x)$ is a fully separable function.

$$\begin{aligned} if|\Delta f_{1,2,\dots,n} - (\Delta f_1 + \Delta f_2 + \dots + \Delta f_n)| < \epsilon \\ then\ x_1, x_2, \dots, x_n\ are\ fully\ separable \end{aligned} \tag{9}$$

However, when eq (8) is not satisfied, we only know that interactions exist in some variable pairs, but we cannot know the interactions exist in which pairs, so we can actively to detect the interactions between variables. Taking the 3-D

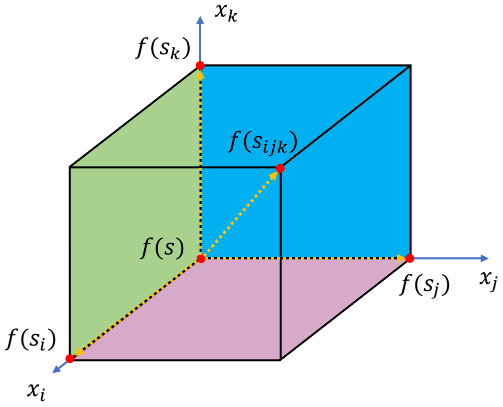

Fig. 5. LINC-R works on 3-D space

space as an example,

$$\begin{aligned}
&if|\Delta f_{ijk} - (\Delta f_i + \Delta f_j + \Delta f_k)| > \epsilon \\
&and\ |\Delta f_{ijk} - (\Delta f_{ij} + \Delta f_k)| < \epsilon \\
&then\ x_i, x_j\ are\ non-separbale \\
&and\ x_k\ is\ separable\ from\ x_i, x_j
\end{aligned}$$

Our target is to apply the heuristic algorithm to find the interactions between all variables as much as possible.

According to the above explanation, in the n-D problem, the objective function in our proposal is designed as eq (10)

$$\min((\Delta f_{1,2,\ldots,n} - \sum^{m}(\Delta f_{i,\ldots,k}))^2) \tag{10}$$

In eq (10), m is the number of sub-problems. Substitute the high-dimensional form of eq (1) into eq (10)

$$\begin{aligned}
&\min((f(s_{1,2,\ldots,n}) - f(s) - \sum^{m}(f(s_{i,\ldots,k}) - f(s)))^2) \\
&= \min((f(s_{1,2,\ldots,n}) + (m-1)f(s) - \sum^{m} f(s_{i,\ldots,k}))^2)
\end{aligned} \tag{11}$$

Eq(11) is the objective function of our proposal.

In our previous work[26], we design a Randomized Local Search (RLS) to optimize this objective function. RLS is simple to be applied and efficient to find a local optimum with powerful exploitation. However, CC framework is sensitive with the decompostion, so in pursuit of high accuracy decomposition and exploration ability of optimizer, we apply the Elitist Reservation GA Algorithm[4], and the flowchart is shown in Fig.6

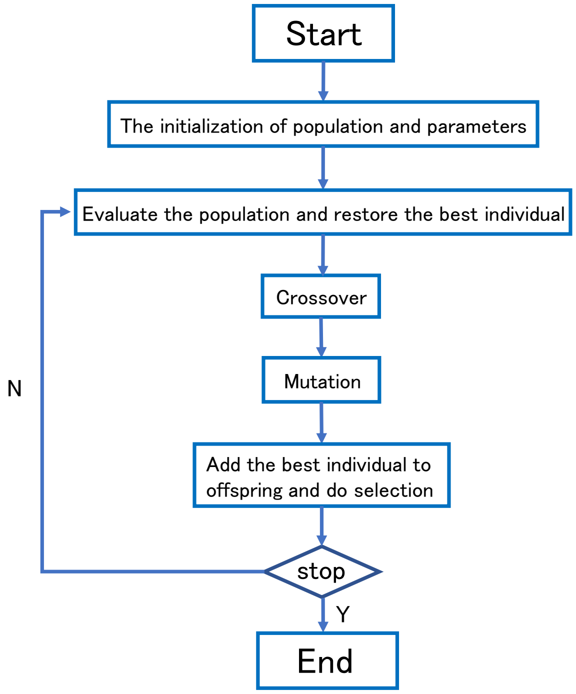
Start
The initialization of population and parameters
Evaluate the population and restore the best individual
Crossover
Mutation
Add the best individual to offspring and do selection
stop
N
Y
End

Fig. 6. The flowchart of the Elitist Reservation Genetic Algorithm

The elitist reservation strategy directly replicates the best individual without crossover, mutation, and selection to the next generation. This strategy can prevent the optimal individual from destroying the superior gene and chromosome structure during crossover mutation.

4 Experiment results and analysis

We evaluate our proposal on the CEC2013 LSGO suite and comparing with DECC-D, DECC-G, and DECC-DG for 25 trail runs. The experiment condition is as follows, Table 1 is the parameters of decomposition optimization, Table 2 is the parameters of sub-problems optimization

Table 1. The parameters of decomposition optimization

Parameter	value
Optimization direction	minimization
Optimizer	Elitist Reservation GA
Population size	20
Max iteration	20
Gene length	8
Stop condition	Objective function<0.1 or reaching max iteration

Table 2. The parameters of sub-problems optimization

Parameter	value
Optimization direction	minimization
Optimizer	DE/current-to-best/1
Fitness evaluation limitation	3,000,000
Population size	30 × Dimension of sub-problem
Scale factor	0.7
Crossover rate	0.9

The convergence curve of the experimental results is shown in Figure 7. In order to confirm the superiority of our proposal, the Kruskal-Wallis test is applied to the fitness at the end of the optimization, if significance exists, then we apply the p-value acquired from the Mann-Whitney U test to do the Holm test. If our proposed method is significantly better than the second-best algorithm, we mark * (significance level 5%) or ** (significance level 10%) in the convergence curve.

And the FEs of DECC-D, DECC-DG and our proposal in decomposition stage is shown in Table 3

It can be seen that on fully separable problems, our proposal can save lots of FEs in the stage of decomposition and acquire the best decomposition, which makes our proposal significantly better than previous methods in f_1 and f_3. In partially non-separable functions, I will explain why our proposal is better than previous methods in two aspects. First, our proposal detects the linearity/non-linearity around the random population instead of DECC-DG which only detects the linearity/non-linearity around a single sample, which makes our algorithm more robust than DECC-DG. In addition, in our decomposition optimization, we apply EGA to optimize, so when a strong interaction between variables (which has a great impact on objective function) is identified, the individual containing the strong interaction is retained as elite to the next generation. We provide a simple example to illustrate, $f(x) = 100(x_1-x_2)^2+(x_2-x_3)^2, s = (0,0,0), s_{123} = (1,1,1)$. In this example, the interaction between x_1 and x_2 is strong, and the interaction between x_2 and x_3 is weak, relatively. We suppose the x_1, x_2, x_3 are

Table 3. The FEs of DECC-D, DECC-DG and our proposal in decomposition stage

	DECC-D	DECC-DG	Proposal
f_1	30000	1001000	5005
f_2	30000	1001000	5005
f_3	30000	1001000	5005
f_4	30000	45694	483515
f_5	30000	668898	483750
f_6	30000	591668	486285
f_7	30000	65992	476370
f_8	30000	72772	482160
f_9	30000	22040	480590
f_{10}	30000	60236	478895
f_{11}	30000	16994	484805
f_{12}	30000	501002	482070
f_{13}	30000	28372	482695
f_{14}	30000	35564	479300
f_{15}	30000	3972	484680

independent from each other in the initialization of decomposition optimization, and the objective function$=(f(s_{123})+2f(s)-(f(s_1)+f(s_2)+f(s_3)))^2 = 40401$. When the interaction only between x_1 and x_2 is identified, objective function$=(f(s_{123})+f(s)-(f(s_{12})+f(s_3)))^2 = 4$, and when only the interaction between x_2 and x_3 is identified, objective function$=(f(s_{123})+f(s)-(f(s_{12})+f(s_3)))^2 = 40000$. It can be seen that when the strong interaction is identified, the improvement to the objective function is greater, the fitness of individual with strong interaction tends to be better, and the probability of well-performed gene structure being preserved to the final solution is also high while some weak interactions are ignored. Although this process increases the error of the sub-problems optimization stage, it can exponentially reduce the search space, which makes it easier to optimize the sub-problems with FEs limitations. Finally, we found that DECC-D and DECC-G are better than both our proposal and DECC-DG in f_{15}. Due to f_{15} being a fully non-separable function, it is completely true to decompose by DECC-DG which directly optimizes f_{15} without decomposition, but it is difficult to optimize the 1000-D problem directly by standard DE. In our proposal, although our algorithm finds some strong interactions between variables, lots of weak interactions still are ignored. How to balance the sub-problems scale and interaction neglect is our research topic in the future.

5 Conclusion

In this paper, we regard the decomposition problem as an optimization problem and design an objective function based on LINC-R. We found that LINC-R can be understood as the additive form of vector and derived to the high-dimensional form. We apply our proposal to the CEC2013 LSGO suite for evaluation and compare our proposal with DECC-D, DECC-G, and DECC-DG. Experimental results show that our proposal is significantly better than the second-best algorithm in most functions. Therefore, decomposition optimization with the LINC-R-based objective function is promising. Although the computational cost of our proposal is better than DECC-DG on fully separable functions, our proposal still requires a large amount of FEs to evaluate the decomposition solutions on non-separable functions, and the computational cost for a solution evaluation is also expensive. So, the application of our proposal to the surrogate model instead of real fitness landscape is a subject of our future research. In addition, decomposition optimization is a combinatorial optimization problem, and designing efficient optimizers is also the center of our study. Besides, the trade-off sub-problems scale and interaction neglect in fully non-separable problems is an interesting topic. Finally, our proposal scales better on higher-dimensional problems than directly using LINC-R or DECC-DG.

6 Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP20K11967.

References

1. Baluja, S.: Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science (1994)
2. Chen, L.J., Ho, Y.H., Hsieh, H.H., Huang, S.T., Lee, H.C., Mahajan, S.: Adf: An anomaly detection framework for large-scale pm2. 5 sensing systems. IEEE Internet of Things Journal **5**(2), 559–570 (2017)
3. Chen, W., Weise, T., Yang, Z., Tang, K.: Large-scale global optimization using cooperative coevolution with variable interaction learning. In: International Conference on Parallel Problem Solving from Nature. pp. 300–309. Springer (2010)
4. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. University of Michigan (1975)
5. Johnston, W.E.: Computational and data grids in large-scale science and engineering. Future Generation Computer Systems **18**(8), 1085–1100 (2002)
6. Köppen, M.: The curse of dimensionality. In: 5th online world conference on soft computing in industrial applications (WSC5). vol. 1, pp. 4–8 (2000)
7. Li, X., Tang, K., Omidvar, M.N., Yang, Z., Qin, K., China, H.: Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. gene **7**(33), 8 (2013)
8. Ling, Y., Li, H., Cao, B.: Cooperative co-evolution with graph-based differential grouping for large scale global optimization. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (2016)
9. Mei, Y., Li, X., Yao, X.: Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems. IEEE Transactions on Evolutionary Computation **18**(3), 435–449 (2014)

10. Munetomo, M.: Linkage identification with epistasis measures considering monotonicity conditions (2002)
11. Munetomo, M., Goldberg, D.E.: Linkage identification by non-monotonicity detection for overlapping functions. Evolutionary computation **7**(4), 377–398 (1999)
12. Omidvar, M., Li, X., Mei, Y., Yao, X.: Cooperative co-evolution with differential grouping for large scale optimization. IEEE Transactions on Evolutionary Computation **18**(3), 378–393 (2014)
13. Omidvar, M., Li, X., Yao, X.: Cooperative co-evolution with delta grouping for large scale non-separable function optimization. pp. 1–8 (2010)
14. Omidvar, M.N., Li, X., Yao, X.: A review of population-based metaheuristics for large-scale black-box global optimization: Part a. IEEE Transactions on Evolutionary Computation pp. 1–1 (2021). https://doi.org/10.1109/TEVC.2021.3130838
15. Omidvar, Mohammad Nabi and Li, Xiaodong and Yao, Xin: A review of population-based metaheuristics for large-scale black-box global optimization: Part b. IEEE Transactions on Evolutionary Computation pp. 1–1 (2021). https://doi.org/10.1109/TEVC.2021.3130835
16. Pelikan, M., Goldberg, D.E., Cantú-Paz, E., et al.: BOA: The Bayesian optimization algorithm. In: Proceedings of the genetic and evolutionary computation conference GECCO-99. vol. 1, pp. 525–532. Citeseer (1999)
17. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A survey of optimization by building and using probabilistic models. Computational optimization and applications **21**(1), 5–20 (2002)
18. Peyré, G., Cuturi, M., et al.: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)
19. Potter, M., De Jong, K.: A cooperative coevolutionary approach to function optimization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **866 LNCS**, 249–257 (1994)
20. Sayed, E., Essam, D., Sarker, R., Elsayed, S.: Decomposition-based evolutionary algorithm for large scale constrained problems. Information Sciences **316**, 457–486 (2015)
21. Sayed, E., Essam, D., Sarker, R.: Dependency identification technique for large scale optimization problems. In: 2012 IEEE Congress on Evolutionary Computation. pp. 1–8 (2012). https://doi.org/10.1109/CEC.2012.6256117
22. Sun, Y., Kirley, M., Halgamuge, S.K.: A recursive decomposition method for large scale continuous optimization. IEEE Transactions on Evolutionary Computation **22**(5), 647–661 (2017)
23. Tan, K., Yang, Y., Goh, C.: A distributed cooperative coevolutionary algorithm for multiobjective optimization. IEEE Transactions on Evolutionary Computation **10**(5), 527–549 (2006)
24. Tezuka, M., Munetomo, M., Akama, K.: Linkage identification by nonlinearity check for real-coded genetic algorithms. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **3103**, 222–233 (2004)
25. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Sciences **178**(15), 2985–2999 (2008)
26. Zhong, R., Munetomo, M.: Random Population-based Decomposition Method by Linkage Identification with Non-linearity Minimization on Graph. In: Transactions on Computational Science & Computational Intelligence. Springer (accepted)

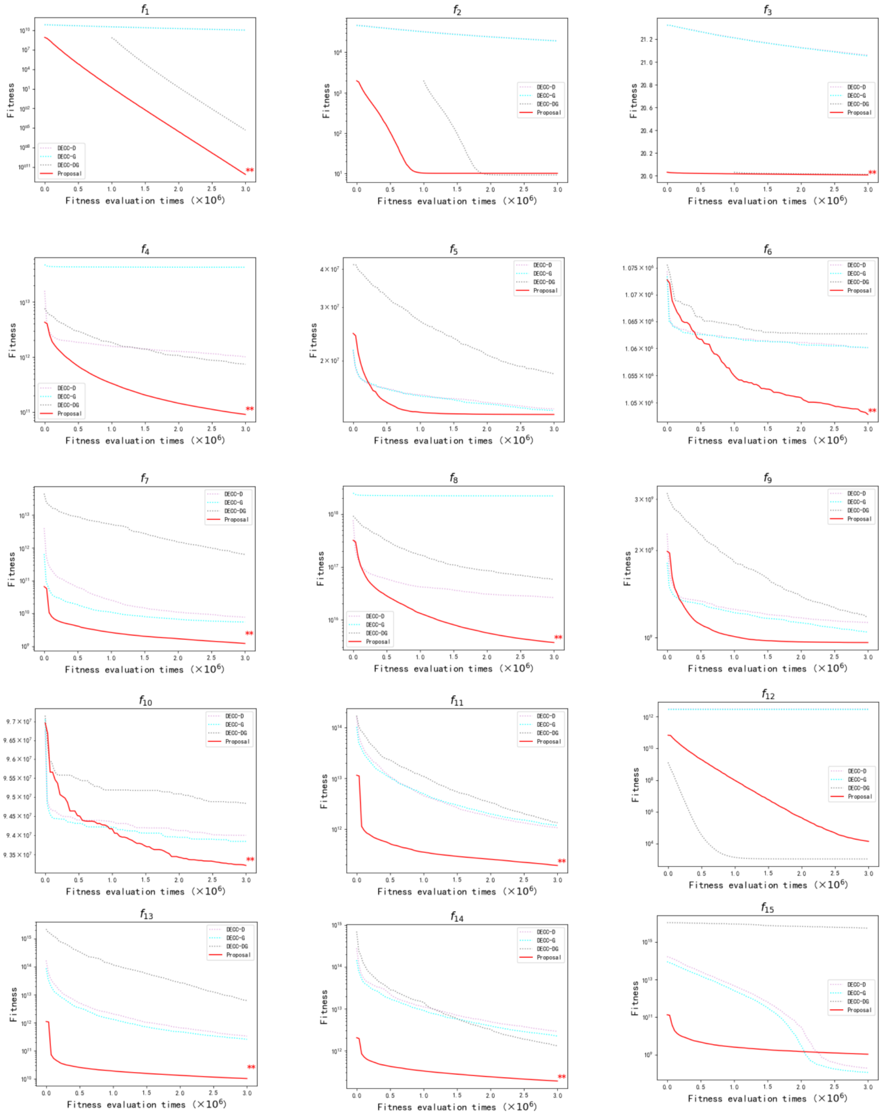

Fig. 7. The convergence curve of DECC-D, DECC-G, DECC-DG, and our proposal. The gaps in the initial period for DECC-D, DECC-DG, and our proposal are FEs consumed in decomposition stage.