Random population-based decomposition method by Linkage identification with non-linearity minimization on graph

Rui Zhong[1] and Masaharu Munetomo[2]

[1] Faculty of Information Science and Technology, Hokkaido University, Sapporo, 060-0814, Japan,
zhongrui19961230@gmail.com,
[2] Information Initiative Center, Hokkaido University, Sapporo 060-811, Japan,
munetomo@iic.hokudai.ac.jp

Abstract. Cooperative co-evolution (CC) has been employed to solve large-scale optimization problems and has achieved exciting results on multiple benchmark suites. In theory, a perfect decomposition method can exponentially decrease the search space without losing optimization accuracy. Therefore, the CC framework has become one of the most popular strategies for solving large-scale optimization problems. However, in practice, it is impossible to design a high-accuracy decomposition method if we have little prior knowledge of the fitness landscape and the high-accuracy decomposition methods often consume a large number of fitness evaluation times(FEs) to detect the interactions between variables. The representative algorithm clustering is the algorithms based on Linkage identification with non-linearity on real code or Differential Grouping(LINC-R/DG). However, the large computational cost and local non-linearity check make these algorithms with limited scalability. In this paper, we propose a novel decomposition algorithm that regards the decomposition problem as an optimization problem and design an objective function based on LINC-R/DG. We mathematically explain the feasibility of our objective function. Numerical experiments show that our decomposition method has broad prospects for solving large-scale problems.

Keywords: Cooperative co-evolution (CC), Large-scale global optimization (LSGO), Linkage identification with non-linearity on real code (LINC-R)

1 Introduction

The CC framework[1] was proposed to solve large-scale optimization problems based on the divide and conquer. The basic idea of CC is to decompose the original problem into multiple non-separable sub-problems, then use evolutionary algorithms (EAs) to solve each sub-problem. This idea has successfully solved large-scale continuous[2], combinatorial[3], constrained[4], and multi-objective[5] problems.

The accompanying problem is the design of the decomposition strategy. Many studies[6–8] have shown that the CC framework is sensitive to problem decomposition strategies, and how to design decomposition strategies has become a hot research topic.

In this paper, we propose a novel decomposition strategy named Random population-based decomposition method by Linkage identification with nonlinearity minimization on the graph (graphLINC-Rmin), our proposal allows an automatic decomposition method that treats the decomposition problem as an optimization problem, an objective function is designed based on LINC on real code (LINC-R).

To illustrate the superiority of our proposal in solving large-scale optimization problems, we tested our proposal on the cec2013 large-scale global optimization (LSGO) suite[9] and comparing with standard Differential Grouping(DG)[2], which shows certain competitiveness.

The rest of the paper is organized as follows. Section 2 describes the gene interaction, we also introduce the CC framework and do a brief survey about current popular decomposition methods. Section 3 introduces our proposal in detail. Section 4 shows the numerical experiment and analyzes the experimental results. Section 5 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 variable decomposition methods.

2.1 Gene interaction

The correlation between genes is known as Epistasis[10] or gene interaction[2]. 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. We extend the concept of Gene interaction to the definition of optimization problems. When $f(x_1, x_2, ..., x_n) = \sum_{i=1}^{m} f(x_{i_1}, ..., x_{i_k}), L_i = \{i_1, ..., i_k\}$ is defined as Linkage set[13], 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, 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, $f(x_1, x_2, ..., x_n) = f(x_1) + f(x_2) + ... + f(x_n)$, 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.

2.2 CC framework and popular decomposition methods

CC framework was first proposed in 1994 based on the idea of divide and conquer, which is known as the main framework for solving large-scale optimization problems. Cooperative Coevolutionary Genetic Algorithm (CCGA)[1] was proposed together with the CC framework. CCGA applies the simplest decomposition method which decomposes n-D problems into $n \times$ 1-D sub-problems. This decomposition method completely ignores the interactions between variables regarded as the pioneer of CC framework. Random decomposition[11] also applies a simple strategy that decompose the original problem to $m \times k$-D subproblems ($n = m \times k$), the hyperparamters are decided by the user. The mathematical explanation shows that in several trial experiments, the probability of two interact variables in the same subcomponent at least once is fairly high.

High-accuracy decomposition is almost impossible with little or no knowledge about the fitness landscape. The Delta grouping[12] alleviates this problem. Delta grouping realizes that when x_i and x_j are separable variables, the difference in coordinates from the initial random population to the optimized population is larger, and when x_i and x_j are non-separable variables, the difference is smaller. Based on this phenomenon, delta grouping 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 dimensional difference. The experimental results on the CEC2008 LSGO suite and the CEC2010 LSGO suite show that this rough estimation method can capture interactions well with little prior knowledge.

In pursuit of higher decomposition accuracy, Linkage Identification by Nonlinearity Check for Real-Coded Genetic Algorithms (LINC-R)[13] are proposed. LINC-R slightly perturbs the two dimensions and identifies the interactions by calculating the difference in fitness. Specifically, this method is shown in eq. (1)

$$
\begin{aligned}
s &= (x_1, x_2, ..., x_n) \\
s_i &= (x_1, ..., x_i + \delta, ..., x_n) \\
s_j &= (x_1, ..., x_j + \delta, ..., x_n) \\
s_{ij} &= (x_1, ..., x_i + \delta, ..., x_j + \delta, ..., x_n) \\
\Delta f_i &= f(s_i) - f(s) \\
\Delta f_j &= f(s_j) - f(s) \\
\Delta f_{ij} &= f(s_{ij}) - f(s) \\
if|\Delta f_{ij} &- (\Delta f_i + \Delta f_j)| < \epsilon \\
then\ x_i\ &and\ x_j\ are\ separable
\end{aligned}
\tag{1}
$$

LINC-R is applied with the random population on the low-dimensional test function and acquires inspired results. Based on the ideas of the LINC-R, Differential grouping (DG)[2] is proposed to apply eq. 1 to the problems up to 1000 dimensions firstly, and the results on the CEC2010 suite show the prospects for solving the large-scale optimization problem. At present, the decomposition method based on LINC-R/DG is considered as a high-accuracy decomposition method, the graphDG[14] even achieves 100% accuracy on the most of benchmark functions in the CEC2010 LSGO suite. However, the disadvantage of DG-based algorithms is that it requires large computational cost to detect the interactions between variables, although several current studies have reduced the computational cost from $O(n^2)$[2] to $O(nlog(n))$ [15], but in much higher-dimensional problems (such as ten thousand, or even millions), the computational cost is completely unacceptable, and due to the high computational cost, when identifying the interactions between variables, the DG-based algorithm often represents the global interaction with the local information around a certain sampling point, which is also needed to be improved.

3 graphLINC-Rmin

In this section, we will introduce the details of our proposal and the techniques. The flowchart of our proposal is shown in fig 1

Next, we will explain our proposal according to the flowchart.

In our proposal, we transform the interactions between variables into a graph and save it as an adjacent matrix. A simple example with 5 variables is shown in figure 2.

Storing the graph as an adjacent matrix allows our adaptive random search to be applied on the matrix. In addition, we randomly generated a population with the size of n in the initialization so that LINC-R can be applied in various samples to overcome the shortage of LINC-R in the large-scale optimization problem.

How to evaluate the decomposition solution is the key to our research. In the past research, a decomposition method of Fitness Difference Minimization[6] assumes the n-D problem can be decomposed into $m \times k$-D sub-problems ($n = m \times k$) and the decomposition problem was regarded as an optimization problem. eq. (2) is the objective function of the optimization problem.

$$min(f(x) - \sum_{i=1}^{m}(f_i(x))^2) \quad f_i(x)\ is\ a\ sub-problem\ with\ s\ variables \tag{2}$$

Fitness Difference Minimization achieves good results on the CEC2010 LSGO suite, however, the assumption of n-D problems can be decomposed into $m \times k$-D sub-problems is not good in most of the problems, although such an operation can greatly reduce the search space of optimization.

In our experiments, we apply the transformation of the LINC-R as the objective function. The detailed explanation is as follows. In LINC-R, We rewrite the eq. 1 to eq. 3

$$\begin{aligned} if|(f(s_{ij}) - f(s_i)) - (f(s_j) - f(s))| < \epsilon \\ then\ x_i\ and\ x_j\ are\ separable \end{aligned} \tag{3}$$

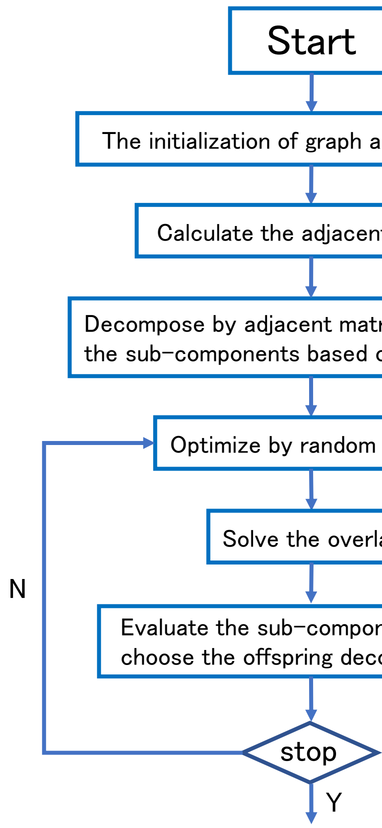


Fig. 1: The flowchart of our proposal

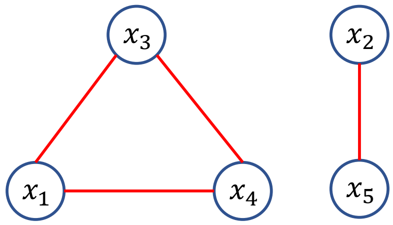
Figure 3 shows how LINC-R works on the separable and non-separable variables. In figure 3, LINC-R compares the difference between different samples with the same perturbation. If the difference is under the threshold ϵ, then LINC-R considers x_i and x_j are separable.

In our proposal, we transform the LINC-R to eq. 4

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

According to figure 4, LINC-R is regarded as the additive form of vectors. If the eq. 4 is satisfied, x_i and x_j are separable variables.

Based on this interesting finding, we next apply the LINC-R to 3-dimension or much higher. In 3-D space, the schematic diagram is shown in figure 5.



(a)

	x_1	x_2	x_3	x_4	x_5
x_1	0	0	1	1	0
x_2	0	0	0	0	1
x_3	1	0	0	1	0
x_4	1	0	1	0	0
x_5	0	1	0	0	0

(b)

Fig. 2: The transformation from the graph to the adjacent matrix. (a) The graph shows the interaction between variables. (b) The adjacent matrix transformed from (a)

Here, we define

$$\begin{aligned} \Delta f_i &= f(s_i) - f(s) \\ \Delta f_j &= f(s_j) - f(s) \\ \Delta f_k &= f(s_k) - f(s) \\ \Delta f_{ijk} &= f(s_{ijk}) - f(s) \end{aligned}$$

So, in 3-D space,

$$\begin{gathered} if |\Delta f_{ijk} - (\Delta f_i + \Delta f_j + \Delta f_k)| < \epsilon \\ then\ x_i, x_j, x_k\ are\ fully\ separable \end{gathered}$$

Therefore, we can reasonably infer that when the dimension reaches n,

$$\begin{gathered} if |\Delta f_{1,2,...,n} - (\Delta f_1 + \Delta f_2 + ... + \Delta f_n)| < \epsilon \\ then\ x_1, x_2, ..., x_n\ are\ fully\ separable \end{gathered} \tag{5}$$

However, when eq. 5 is not satisfied, we only know that interactions exist in some variables, but we cannot know the interactions exist in which pair(s) of variables, but we can actively to detect the interactions between variables. Taking the 3-D space as an example,

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

According to the above illustration, in the n-D problem, the objective function in our proposal is defined as eq. 6

$$min((\Delta f_{1,2,...,n} - \sum_{i,j,...}^{m} (\Delta f_{i,j,...}))^2) \tag{6}$$

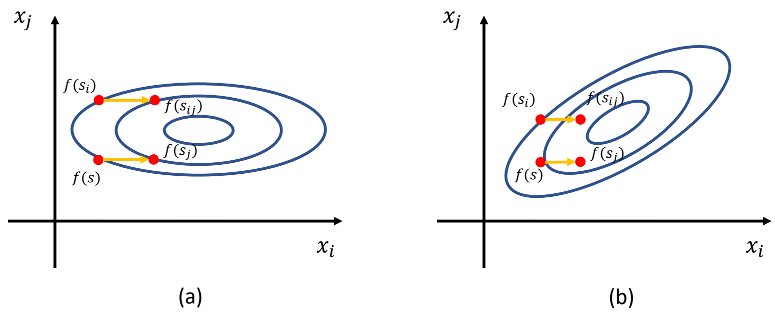


Fig. 3: (a) LINC-R works on the separable variables. (a) LINC-R works on the non-separable variables.

In eq. 6, m is the number of sub-components. And we apply eq. 1 to eq. 6

$$\begin{aligned} &min((f(s_{1,2,\ldots,n}) - f(s) - \sum_{i,j,\ldots}^{m}(f(s_{i,j,\ldots}) - f(s)))^2) \\ &= min((f(s_{1,2,\ldots,n}) + (m-1)f(s) - \sum_{i,j,\ldots}^{m} f(s_{i,j,\ldots}))^2) \end{aligned} \tag{7}$$

eq. 7 is the objective function of our proposal.

We design an adaptive random search algorithm based on the adjacent matrix of the graph. The basic idea is that in the early stage of the algorithm, the algorithm finds an acceptable decomposition result by changing a large number of connections states on graph, then in the late stage of the algorithm, the algorithm modifies the a little number connection states, and the algorithm gradually reaches convergence. The pseudocode for adaptive random search is shown in algorithm 1.

In the solution found by adaptive random search, there is a high probability of existing overlapping variables, and how to deal with overlapping variables is also a common problem. Here, we apply a simple strategy to randomly ignore the interactions in overlap variables.

4 Experiment results and analysis

We evaluate our proposal on the CEC2013 LSGO suite and compare it with the standard DG for 25 trail runs. The experiment condition is as follows, Table 1

Algorithm 1 Random search for decomposition optimization

Input: *Parameters*: adj_matrix, cur_iter, max_iter
Output: adj_matrix

```
function RandomSearch(Parameters)
    adjust_connection_size ← Adaptive(cur_iter, max_iter) // decide the search scale adaptively by current iteration and max iteration
    i ← 0
    points ← list // save the adjust connections
    while i < adjust_connection_size do
        point ← [a, b] // a != b
        if point not in points then
            points ← point
            i++
        end if
    end while
    for point in points do // reverse the original connections
        if adj_matrix[point[0]][point[1]] == 1 then
            adj_matrix[point[0]][point[1]] = 0
            adj_matrix[point[1]][point[0]] = 0
        else
            adj_matrix[point[0]][point[1]] = 1
            adj_matrix[point[1]][point[0]] = 1
        end if
    end for
    return adj_matrix
end function
```

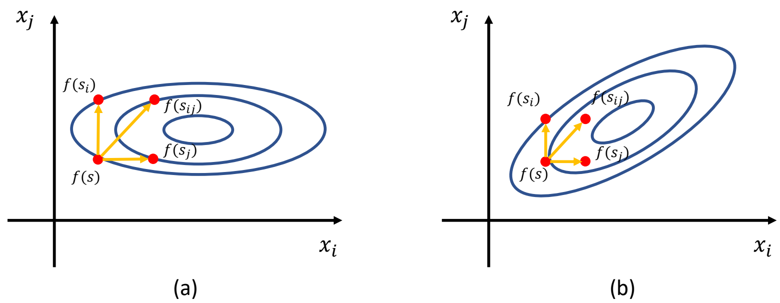


Fig. 4: (a) The transformed LINC-R works on the separable variables. (b) The transformed LINC-R works on the non-separable variables

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
Initial connection matrix	Zero-like
Population size	5
Search adaptive strategy	Linearity
Overlap connection ignore rate	0.5
Max iteration	100
Stop condition	Objective function<0.1 or reaching max iteration

To evaluate the superiority of our proposal, we apply the Mann-Whitney U test between our proposal and standard DG at the end of the sub-problem optimization stage. And we mark '*' when our proposal is better than standard DG with $p < 0.05$, '**' when our proposal is better than standard DG with $p < 0.01$. The convergence curve is shown in figure 6

In the most of benchmark functions in the CEC2013 LSGO suite, the performance of our proposal is significantly better than standard DG.

In Table 3, we show the required FEs of decomposition in standard DG and our proposal on the CEC2013 LSGO suite.

From the table 3, our proposal can save an amount of FEs in the stage of decomposition on the fully separable functions and allocate the rest FEs to the sub-problems optimization stage, which makes our proposal significantly better

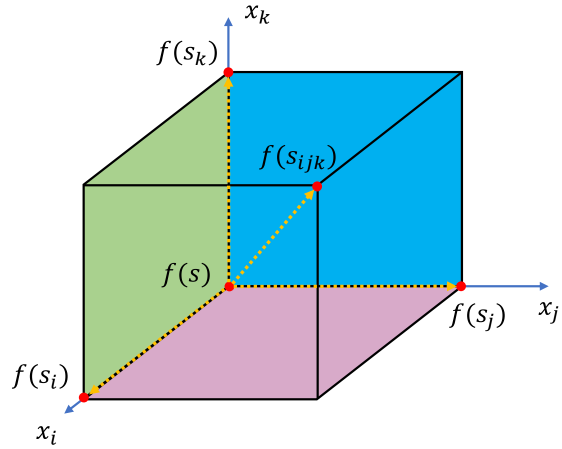


Fig. 5: LINC-R applied in 3-D space

than standard DG. In partially non-separable functions, I will explain why our proposal is better than standard DG in two aspects. First, the objective function in our proposal is based on the multiple samples instead of DG only detecting the linearity/non-linearity around a single sample to represent the interactions of variables in the global search space, which makes our proposal more robust. Furthermore, in our decomposition optimization stage, strong interactions tend to be easier to restore because it contributes more improvement to reducing the objective function than weak interactions. 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 independent

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-Fitness Evaluation times(FEs) in decomposition
Population size	30 $\times$ Dimension of sub-problem
Scale factor	0.7
Crossover rate	0.9

Table 3: The required FEs in decomposition of our proposal and standard DG

Function	Standard DG	Proposal
f_1	1001000	5000
f_2	1001000	5000
f_3	1001000	5000
f_4	45694	254290
f_5	668898	291075
f_6	591668	264775
f_7	65992	304625
f_8	72772	278230
f_9	22040	266230
f_{10}	60236	269470
f_{11}	16994	292635
f_{12}	501002	259860
f_{13}	28372	274320
f_{14}	35564	276295
f_{15}	3972	261225

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 than the weak interaction, so, in our adaptive random search, when the strong interactions are identified, there is more probability of retaining the strong interaction as the current optimum decomposition result 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.

5 Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP20K11967.

6 Conclusion

In this paper, we regard the decomposition problem as an optimization problem and design a novel decomposition algorithm based on high dimensional LINC-R. In our research, we found that LINC-R can be understood as an additive form of vector and reasonably derived to a high-dimensional form as the objective function in our proposal. To evaluate our proposal, we apply our proposal on the

CEC2013 LSGO suite and comparing with the standard DG. The experimental results show that the performance of our proposal is significantly better than standard DG in most test functions, and the computational cost is also better than standard DG in fully separable and some partially non-separable functions, so high dimensional LINC-R applied as an objective function for decomposition optimization is a promising research topic. However, due to the limitations of the random search algorithm, it is often impossible to find the local optimum in decomposition optimization, and in high-dimensional problems, such as Genetic Algorithms, the coding of chromosomes will become sparse and long, so it is difficult to apply traditional Evolutionary Algorithms to optimize the decomposition problems, therefore, how to optimize the decomposition problem is a challenge of our future research. Furthermore, our proposal scales better on higher-dimensional problems than directly using LINC-R on the problem, so the application of our proposal to solve much higher-dimensional problems is also a theme of our research.

References

1. M. Potter and K. De Jong, "A cooperative coevolutionary approach to function optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 866 LNCS, pp. 249–257, 1994.
2. M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.
3. Y. Mei, X. Li, and X. Yao, "Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 435–449, 2014.
4. E. Sayed, D. Essam, R. Sarker, and S. Elsayed, "Decomposition-based evolutionary algorithm for large scale constrained problems," *Information Sciences*, vol. 316, pp. 457–486, 2015.
5. K. Tan, Y. Yang, and C. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527–549, 2006.
6. E. Sayed, D. Essam, and R. Sarker, "Dependency identification technique for large scale optimization problems," in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.
7. M. N. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization: Part a," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2021.
8. 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.
9. X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," *gene*, vol. 7, no. 33, p. 8, 2013.
10. M. Munetomo, "Linkage identification with epistasis measures considering monotonicity conditions," 2002.

11. Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.
12. M. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," 2010, pp. 1–8.
13. M. Tezuka, M. Munetomo, and K. Akama, "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)*, vol. 3103, pp. 222–233, 2004.
14. Y. Ling, H. Li, and B. Cao, "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.
15. Y. Sun, M. Kirley, and S. K. Halgamuge, "A recursive decomposition method for large scale continuous optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 647–661, 2017.

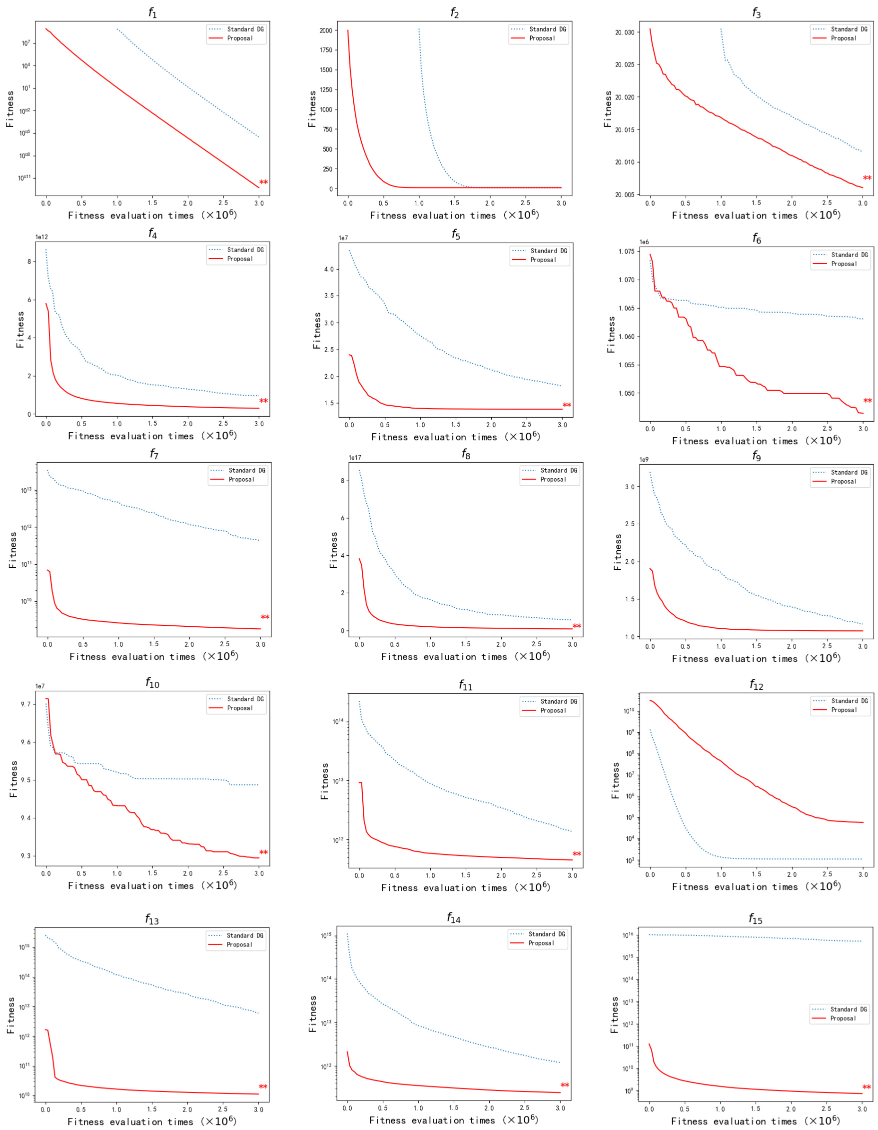


Fig. 6: The convergence curve of our proposal and standard DG