We would like to thank the reviewers for their valuable comments and suggestions. Based on these comments, we have revised our manuscript. The main revisions and responses are as follows:

Revisions Based on the Comments of Reviewer #1

1. English is OK, but it is also suggested to be corrected by a native speaker.

Answer 1: Thank you for reading our manuscript carefully. We have checked the entire manuscript and refined the language.

2: The titles of x-axis and y-axis in Fig. 1 need to be added.

Answer 2: Thank you for your suggestion. Captions have been added to Figure 1, with the x-axis representing the dimension and the y-axis representing the improvement of the corresponding dimensions.

3: In the experimental study, the proposed algorithm should be compared with other decomposition methods.

Answer 3: Thank you for your valuable suggestions. We have added a new comparison with other four well-known or state-of-the-art algorithms. The comparison results are listed in Table III and analysis of the experiment result are presented.

Revisions Based on the Comments of Reviewer #2

1. Comprehensively represent existing works for separable and non-separable LSGO problems.

2. Clearly describe the idea of the proposed algorithm.

Answer 1-2: Thank you so much for your compliment.

3.1: In Algorithm 1, it is described as "Initialize the neighborhood radius(ϵ) of DBSCAN and minimum points (MinPts)", but how to configure these parameters for DBSCAN in practice is unclear.

Answer 3.1: Thank you for your suggestion. We have added the specification of the DBSCAN configuration neighborhood radius (ϵ) and minimum points (MinPts) in Algorithm 1. The neighborhood radius is set to 10^-12 and the minimum points is set to 10. These two parameters is set to a small value because the improvement of decision variables are highly clustered.

3.2: You use four strategies to decompose the problem, and optimize each sub-group separately. If there are enough evaluation times for each sub-problem, because comparing to use one strategy, there are more sub-groups in your algorithm.

Answer 3.2: For fair comparison, the maximum number of function evaluations (FEs) is set to 3×10^6 for all algorithms for large-scale problems. Although we decompose the problem into subproblems (decompose decision variables into sub-groups), when optimizing the variables in one subgroup, the variables in other subgroups are kept fixed in their current best values. And the function evaluations are made with all the decision variables combined together. Thus, the total number of function evaluations is the same as other non-decomposition algorithms, it is fair comparison.

4.1: The dataset for experiment only contains 6 problems, which is relatively little to represent the generalization of your algorithm. Besides fully non-separable problems, maybe separable problems can be included, because it is reasonable to expect that a decomposition method works on separable problems.

Answer 4.1: For separable problems, there are already abundant decomposition methods existed while little attention is paid to the decomposition of fully non-separable problem. Thus in this paper we focus on fully non-separable problems. For separable problems, decomposition accuracy is one way to verify the effectiveness of the decomposition method while for fully non-separable problems, there is no criteria for decomposition quality since theoretically it should not be decomposed because all the decision variables are interacted with each other. So in this paper we want to show that by reasonably decomposing fully non-separable problems, the optimization can gain better results.

4.2: And you only compare your decomposition algorithm with a non-decomposition version to verify the effectiveness of the decomposition scheme, which can be regraded as an ablation study. You don't compare your method with other existing algorithms work for non-separable LSGO, like those mentioned in your related work, to show the advancement of the proposed algorithm in this paper.

Answer 4.2: Thank you for the advice. We have added one more comparison in the experimental section, which presents comparative results and analysis with two well-known algorithms and two state-of-the-art large-scale algorithms. Comparison result further demonstrate the effectiveness and advantage of the proposed method.

5.1: The paper needs to be carefully proofread, because there are some mistakes about spelling. For example, 'weather' in ' weather or not should we decompose …' locating at the last paragraph of Sec 2.

Answer 5.1: Thank you for your advice. We have double-checked the manuscript and corrected some spelling mistakes and refined the language.

5.2: Please delete the 1st, 2nd, 3rd before the name of the authors.

Answer 5.2: These symbols are from the latex template, we will delete them if required in future version.

Revisions Based on the Comments of Reviewer #3

1: The explanation of motivation and contribution is not sufficient in the introduction of this paper. Only decomposition strategy, fully separable problem and partially separable problem are presented in the introduction. However, this paper is oriented to the decomposition of fully non-separable problem. So, a comprehensive discussion on the nature of fully non-separable problem, the rationale behind decomposing such problems (e.g., for enhancing efficiency), the challenges encountered by existing methods, and the specific contributions offered by the method proposed in this study, shoule be included.

Answer 1: Thank you for your suggestion. The introduction section has been thoroughly rewritten for better explanation of the problem and motivation. Moreover, two examples (Equation 1 and 2) are given for detail explanation of fully non-separable problems. We also added the contribution of this paper in the last paragraph of introduction section.

2: The related work section is too long and lack of focus. Consider trimming the content and relocating some of the information regarding the problem decomposition method to the introduction section.

Answer 2: Thank you for your suggestion. We have rewritten the related work section, reduced some unimportant content and refined the content to make it more readable.

3: In the experimental results section, the proposed method seems to have a very good performance. However, there remain certain aspects that require further elucidation. Firstly, regarding the setting of the maximum fitness evaluation times, for the decomposed problem whether the maximum FEs is applied to each group separately or the sum of the evaluations of all groups is the maximum FEs. If it is the first, whether this will bring the unfairness of comparison. Secondly, a more comprehensive elucidation of the optimization performance after problem decomposition is needed. The paper has repeatedly emphasized that the fully non-separable problem should not be decomposed. Because this may cause the optimization to inevitably fall into local optima. If decomposition is deemed necessary, there needs to be a delicate balance between reducing optimization difficulty through decomposition and disrupting variable connections.

Answer 3.1:

The maximum FEs is the sum of the evaluations of all sub-groups. For fair comparison, the maximum number of FEs is set to 3×10^6 for all algorithms for large-scale problems. Although we decompose the problem, when optimizing the variables in one subgroup, the variables in other subgroups are kept fixed in their current best values. And the function evaluations are made with the all the decision variables combined together. Thus, the total number of function evaluations is the same as other non-decomposition algorithms, it is fair comparison.

Answer 3.2:

This is a good question, we did make trade-offs. Since optimizing the large-scale problem as a whole is very ineffective and inefficient, so we decide to decompose the problem although it should not since all variables are interacted. So our approach offers multiple decomposition strategies for these problems in order to increase the diversity of the decomposition. Specifically, in this paper we provide more than 20 different decomposition results for fully non-separable problems to make the trade-off. In this way, variables with strong interact can have more chance to be grouped together. The experiments verified that this method is effective.

4: The use of some symbols needs to be checked. For example, "e-2", "e^10" in the second paragraph of the introduction. The author seems to mean "10^-2" or "1e-2"? Some language disorders and word errors need to be checked. For example, in the last paragraph of page 4, it should be “fully non-separable large problem”?

Answer 4: Thank you for your suggestion. We have standardized the notation in this manuscript to a format as 10^-2 and 10^10. We also checked the manuscript carefully and refined the language.

A new decomposition method for fully non-separable large-scale problem

1st Wenlong Song
School of Computer Science and Technology
Xi'an University of Posts and Telecommunications
Xi'an, China
fangtang_s@163.com

2nd Haiyan Liu
School of Computer Science and Technology
Xi'an University of Posts and Telecommunications
Xi'an, China
hyliu83@126.com

3rd Yi Cheng
School of Computer Science and Technology
Xi'an University of Posts and Telecommunications
Xi'an, China
cyjy2000@163.com

4thYuping Wang
School of Computer Science and Technology
Xidian University
Xi'an, China
ywang@xidian.edu.cn

***Abstract*—Large-scale global optimization (LSGO) problem is such a problem containing hundreds or thousands of decision variables. Solving LSGO problems is very challenging due to its high nonlinearity, high dimensionality and too many local optimal solutions. LSGO algorithms typically employ a divide-and-conquer strategy to solve the problem. This involves decomposing the LSGO problem into subproblems and solving them individually. The most challenging LSGO problems are those that cannot be divided due to interactions between variables. We propose a decomposition method to divide variables of fully non-separable LSGO problems into sub groups in a reasonable way. We design a hybrid decomposition scheme by integrating four decomposition strategies so as to get better decomposition result. Numerical experimental results show that the proposed decomposition method is effective and totally outperforms the correct decomposition (non-decomposition).**

***Index Terms*—large-scale optimization; decomposition/grouping method; dimension reduction; non-separable problem**

I. Introduction

Many optimization problems appeared in scientific and engineering applications often need to deal with a large number of decision variables, such as in network scheduling, artificial intelligence model training and complex economic decision-making. These optimization scenarios often entail hundreds or even thousands of decision variables, constituting what is commonly referred to as LSGO problems, a burgeoning area of research interest in recent years.

LSGO problem is a kind of very complicated and hard problems. For example, when considering the CEC'2013 large-scale benchmark suite comprising 15 test problems, the latest algorithms can only manage to attain fewer than five global optimal solutions with an accuracy error below 10^{-2}. Conversely, the remaining test problems exhibit significantly larger accuracy errors, reaching magnitudes as high as 10^{10}. Therefore, LSGO is still far from being solved and more research and effort need to be invested in this problem.

The extreme difficulty of the problem is mainly due to three key reasons. Firstly, the problem's high dimensionality, with up to 1000 decision variables, leads to an enormous search space. Thus, for most optimization algorithms, it is impossible to conduct a deep search, resulting in a shallow exploration dilemma. Secondly, the problem's unknown landscape, irregularity, oscillation, and ill-conditioning pose significant challenges to the optimization process. Additionally, large-scale problems often have numerous local optimal solutions that can hinder the optimization algorithms' progress towards the global optimal solution.

When dealing with LSGO problems, it is advisable to use the divide-and-conquer approach. This involves breaking down the problem into smaller sub-problems and solving them separately. One effective framework for this is the cooperative co-evolution (CC) framework [1]. The CC framework works by decomposing a large-scale problem into smaller-scale sub-problems and then optimizing each sub-problem individually using evolutionary algorithms. To evaluate the fitness of individuals in each sub-problem, they are combined with the best individuals from all other sub-problems to form a complete solution.

The divide and conquer strategy is effective when large-scale problems can be well decomposed. However, the interdependencies or interactions among variables of the large-scale problem make it hard to decompose effectively and efficiently. To enhance optimization outcomes, it is imperative to optimize these interacting variables as a whole. Conversely, an incorrect decomposition of interacting variables can lead to unsatisfactory optimization results within the CC framework.

In the context of large-scale problems, separable and non-separable concepts is commonly utilized to characterize variable interactions. Separability refers to the independence of certain variable(s) from others, while non-separability indi-

National Natural Science Foundation of China (No. 62002289).

cates the presence of variable interactions within the problem. A problem with 'n' decision variables is deemed fully non-separable when all variables are interconnected with each other. On the other hand, a separable problem denotes scenarios where some variables are independent with other (group of) variables. In other words, the decision variables can be divided into sub-groups. The variables within one sub-group interact with each other while independent with variables from other sub-groups. Fully non-separable problems, on the other hand, could not be decomposed conventionally since all decision variables are interacted with each other.

To better illustrate separable and fully non-separable problems, we give two simple examples with only four variables in equation 1 and 2.

$$f_1(x) = x_1 * x_2 + x_3 * x_4 \tag{1}$$

$$f_2(x) = (x_1 * x_2 + x_3 * x_4)^2 \tag{2}$$

Since the four variables of x_1, x_2, x_3 and x_4 in equation 1 can be divided into two sub-groups $\{x_1, x_2\}$ and $\{x_3, x_4\}$ with variables in different sub-groups have no interactions, f_1 is separable problem. Conversely, the four variables in equation 2 can not be divided because all these four variables interact with each other, thus f_2 is fully non-separable problems. Thus, fully non-separable problems are the most challenging large-scale problem because they can not be divided and conquered using CC framework.

One critical challenge in CC algorithms is how to correctly and effectively identify variable interactions to ensure accurate decomposition of the large-scale problem. Multiple innovative approaches have been proposed to address this issue. The differential grouping (DG) [2], global differential grouping strategy (GDG) [3], DG2 [4], fast interdependency identification(FII) [5], efficient recursive differential grouping method (ERDG) [6] and merged differential grouping method (MDG) [7] are some of them. However, these methods can only detect and decompose separable problems. For fully non-separable large-scale problems,effective decomposition methods are needed to reduce the dimensionality and thus to reduce the difficulty of the problem.

We noticed that the high dimensionality is the most difficult part for solving LSGO problems, especially for fully non-separable large-scale problems. The extremely large search space makes LSGO algorithms ineffective and impossible to conduct a thorough search. By decomposing the problem, the dimensionality and search space can be greatly reduced, thus the efficiency of optimization algorithms can be greatly improved. In this paper, we focus on solving fully non-separable LSGO problems by decomposing the problem and reducing dimensionality and the search space since for separable problems, there have been already a lot of decomposition algorithms.

II. Related work

In recent years, LSGO has become a research hot spot and many algorithms have been designed to solve it. Some early work of LSGO include a variety of CC algorithms [2], [4], [5], [8]–[11], distribution estimation algorithms [12]–[15], memetic algorithms [16]–[18] and hybrid algorithms [19], [20]. Large-scale optimization algorithms are commonly categorized into two main groups: decomposition-based algorithms, often referred to as CC algorithms, and non-decomposition-based algorithms.

CC algorithms are intended to divide large-scale problems into smaller sub-problems or sub-groups, which are then solved individually using evolutionary algorithms. A critical challenge in CC algorithms lies in effectively identifying variable interactions to ensure accurate decomposition of the large-scale problem. Several innovative approaches have been proposed to address this issue. Notable methods include Delta grouping [21], correlation identification grouping [22], and variable interaction learning [23], [24], which focus on measuring the impact of variable changes on function values to facilitate decomposition. The DG [2] technique is the first method capable of systematically and automatically decomposing large-scale problems with high accuracy. DG identifies pair-wise variable interactions by examining changes in function values and thus is computationally intensive and limited to direct variable interactions. To overcome these limitations, enhanced versions of the DG method have been developed. For instance, Mei et al. introduced the GDG [3] to identify complete variable interactions. Additionally, DG2 [4], proposed by Omidvar *et al.*, improves the efficiency and accuracy of differential grouping with reduced computational costs. Efforts to further enhance the efficiency of decomposition methods have led to the introduction of innovative approaches such as the FII [5] and the recursive differential grouping method (RDG) [25]. FII compares the current variable to all remaining variables to reduce computational costs and RDG uses a bisection identification to gain efficiency. Building upon these advancements, researchers have introduced more resource-efficient techniques like the ERDG [6] and the MDG [7].

In the realm of large-scale optimization, various decomposition strategies have been explored to break down the complex problems into smaller and more manageable subproblems. Fan *et al.* [26] introduced a kernel fuzzy C-means clustering approach for this purpose. In another study [27] utilized a k-means clustering method to group variables based on their main effects on function values. Ma *et al.* [28]put forth a localized control variable analysis method that involves categorizing decision variables using guiding reference vectors. Despite these efforts, the precision of these decomposition techniques falls short compared to the differential grouping methodology and its derivatives.

While some researchers treat large-scale problems as black-box, while some of them treat it as white-box because many real-world problems come with explicit expressions that can provide valuable information for more accurate decomposition. The existing literature [29] proposed a formula-based grouping (FBG) strategy that leverages problem-specific information to identify variable interactions through pattern recognition.

However, similar to other grouping methods, FBG can not decompose fully non-separable problems. To address this limitation, Liu *et al.* [30] introduced a contribution-based decomposition (CBD) method capable of efficiently handling large-scale fully non-separable problems. Nevertheless, CBD's fixed group size approach may yield suboptimal results. In response, in [31] authors proposed a hybrid deep grouping (HDG) method that not only considers variable interactions but also evaluates the essentialness of variables. HDG facilitates profound decomposition of large-scale problems by selectively eliminating trivial variables from subproblems. Nonetheless, its effectiveness in capturing variable interactions within fully non-separable problems is limited. In a another study, Liu *et al.* [32] introduced a space reduction based algorithm that employs a multi-grouping strategy tailored for addressing fully non-separable large-scale problems. This method shed some new insight in decomposing fully non-separable problems.

Inspired by particle swarm optimization (PSO), Cheng *et al.* [33] developed the competitive swarm optimizer (CSO) employing pairwise competition and novel update strategies for large-scale problem solving. Literature [34] introduced a level-based learning swarm optimizer (LLSO), which categorizes particles into distinct levels based on their fitness values to maintain a balance between exploitation and exploration during updates. In [35] a reinforcement level-based PSO algorithm (RLLPSO) is introduced. RLLPSO aims to improve population diversity and enhance algorithm efficiency by integrating reinforcement learning techniques and level competition mechanisms. Deng *et al.* [36] proposed a ranking-based biased learning swarm optimizer (RBLSO) to solve LSGO problems in which two learning strategies are used to increase diversity. An adaptive granularity learning distributed PSO (AGLDPSO) is proposed in [37] to enhance search efficiency and convergence speed. By dynamically adapting learning granularity based on the search state, the algorithm achieves a balance between global exploration and local optimization.

Despite the numerous algorithms developed for LSGO, the results remain unsatisfactory. The primary challenge lies in the high dimensionality, extensive search space and the numerous value combinations of decision variables for sampling. Particularly, for fully non-separable LSGO problems, the decision of whether to decompose them and how to decompose them is an open problem. While theoretically these problems should not be decomposed due to intricate variable interactions, existing algorithms struggle to efficiently handle these problems. Therefore, we address this issue by proposing a dedicated decomposition method tailored for fully non-separable problems. We believe that trade-offs should be made to balance the non-separability characteristics and the reduction of dimensionality and search space of fully non-separable problems.

III. Methodology

The CC framework is widely used for tackling large-scale optimization problems, provided that the problem can be appropriately decomposed. However, in the case of fully non-separable large-scale problems where all variables exhibit interactions, the conventional notion of 'correct' decomposition, involving the segregation of variables, is not applicable. In such scenarios, the optimal approach would be to treat all variables as a whole for optimization and decomposition should not be made. However, the high dimensionality and expansive search space of these problems make such a strategy inefficient and ineffective. Therefore, the challenge is to make a balance between the notion of 'correct' decomposition and effective decomposition.

Thus, in this paper we try to design some reasonable decomposition strategies for the trade-off to make the optimization more effective and efficient.

A. A new decomposition method of fully non-separable large-scale problem

Literature [30] first presents the idea of decomposing fully non-separable problems and the CBD method is proposed to make the decomposition. However, this grouping method has some drawbacks: firstly, the group size is fixed and thus is not suitable for various problems; Secondly, this grouping method only provides a single decomposition result, which may not be appropriate for fully non-separable problems where all variables interact with each other. Thus if we can design a decomposition method that can provide a series of different decomposition results, that will better fit for the characteristic of fully non-separable problem. Enlighted by this, we design a new decomposition method for fully non-separable problem named k-means density-based(KDB) clustering method.

Firstly, we adopt the self-adaptive discrete scan method in literature [30] as our line search method to gain the improvements of all the variables. In this paper, the self-adaptive discrete scan method is executed for 10 times and the improvement value will be stored in a matrix named F_m. We analyzed and noticed that the distribution of the improvement matrix F_m is uneven. At the beginning of the optimization process, the improvement is significant while in the late stages, the improvement drops dramatically. Only a few dimensions can maintain a high improvement value while the improvement values of most dimensions are in the order of 10^{-4} magnitude or even smaller. Fig 1 shows that the distribution of improvements, X-axis is the 1000 variables of the problem and Y-axis presents the improvement. We can see from Fig 1 that the improvement values mainly clustered around the 0 coordinate which makes it difficult to group them well.

Our goal is to decompose the variables so that the ones with similar improvement level will be grouped together while maintain some diversity of the group at the same time. To accomplish this objective, we have developed a novel decomposition technique for fully non-separable problems that can generate over 20 distinct decompositions. In this case, we can keep a good balance and make trade-offs between the non-separable characteristics and decomposition. In the proposed

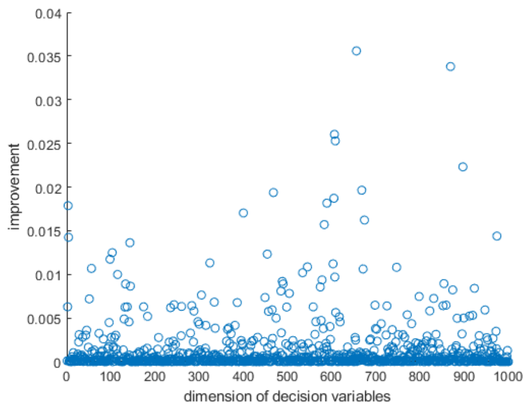

Fig 1. Distribution of the improvement of all variables

decomposition method, four decomposition strategies are designed :

1) A cluster based decomposition method named KDB clustering
2) A cluster based decomposition method named KDB clustering variant
3) A correlation analysis based decomposition method
4) A random grouping strategy for decomposition

For the cluster based decomposition methods, k-means [38] and DBSCAN [39] cluster algorithms are adopted. These two clustering algorithms have different clustering strategies and we can use them to divide data into different clusters with each cluster contains data at the same level. However, these two cluster algorithms can not be applied directly. K-means is very sensitive to the selection of initial points and is not capable of handling noise and clusters with non-convex shapes, thus k-means tends to see dimensions with big improvement value as noise and ignore them. DBSCAN's performance in processing high-dimensional data may decrease, and its ability to handle complex clusters is limited, resulting in a significant increase in computational complexity. To tackle these challenges, we design a new cluster based decomposition method named KDB clustering. We first use the elbow method to determine the initial number of clusters in k-means. Then, we apply DBSCAN to only a portion of the clusters to reduce computational costs.

Specifically, the main steps of KDB clustering method are as follows:

- Step1: Data preprocessing: initialize the number of clusters, extract the rows and dimensions of the improved matrix, in order to prepare for the subsequent cluster analysis.
- Step2: K-means clustering: Using k-means algorithm to cluster the pre-processed data. K-means algorithm divides the data by minimizing the square error within the cluster, and is good at dealing with clusters with convex shapes.
- Step3: Low-density cluster recognition: The clustering results of k-means are analyzed to identify low-density clusters (or noise points), that is, those clusters that are not accurately divided by k-means clustering algorithm.
- Step4: DBSCAN clustering: Further clustering of low-density clusters (or noise points) is performed using DBSCAN algorithm. DBSCAN algorithm can effectively deal with noise points and non-convex clusters, and has a good effect on these difficult cases.
- Step5: Clustering result merging: The clustering results of k-means and DBSCAN are merged to obtain the final clustering result.

The pseudocode of KDB clustering method is presented in Algorithm 1.

Algorithm 1 Pseudo code of KDB clustering method

Require: Improvement matrix F_{mat}, dimension D;
Ensure: Grouping results $group_i$, average improvement of each group $improve_i$;

Take out the number of rows of the improvement matrix $rows$;
for i=1:rows **do**
  Extract the i-th corresponding data and dimension D of F_{mat};
  The number of clusters $K \in [10, 20]$ is generated randomly;
  The improvement, dimension D and K are used by k-means to generate clustering;
  for m=1:K **do**
    The distance between data points in each cluster is calculated, and the average distance is used as the density measure;
    Mark the group with less than the average density as $index$;
  end for
  for n=1:index **do**
    Initialize the neighborhood radius($\epsilon = 10^{-12}$) of DBSCAN and minimum points (MinPts = 10);
    Call the DBSCAN clustering function to cluster the low-density group;
  end for
  The clustering results of k-means and DBSCAN were merged to get the grouping result $group_i$ and the average improvement of each group $improve_i$ are recorded;
end for

The main idea is to use k-means for clustering first, and then use DBSCAN to further partition and optimize the grouping of low density levels from the clustering results of k-means. This combined method can make full use of the respective advantages of k-means and DBSCAN, so as to make up for their shortcomings and obtain more comprehensive and accurate clustering results.

It is important to note that in k-means, the goal is to obtain as many diverse subgroups as possible. The optimal number of clusters K, according to the elbow rule test, is between 10 and 20. Therefore, we randomly select a value for K within this range to ensure diversity among the subgroups. Moreover, given the concentrated distribution of the improvement of decision variables, our aim is to cluster variables exhibiting similar improvement into same groups. To avoid the scenario where all variables are clustered together in a single subgroup, we opt for smaller parameter values. Specifically, we set the neighborhood radius ϵ and MinPts at 10^{-12} and 10 respectively to facilitate effective variable grouping while maintaining subgroup diversity.

Clustering and grouping are a relatively flexible process, different sequences will highlight different data characteristics, so as to get different clustering results. In order to decompose fully separable large-scale problems and obtain various decomposition results, it is reasonable to divide the interacting variables into a group so as to simplify the problem and make the optimization easier.

On this basis, we designed another variant of KDB Clustering, in which we first use DBSCAN to perform on the raw data analysis, and then use k-means to further handle the data that do not meet the requirements. The pseudocode of KDB clustering variant is presented in Algorithm 2.

Algorithm 2 Pseudo code of KDB clustering variant

Require: Improvement matrix F_{mat};
Ensure: Grouping results $group_i$, average improvement of each group $improve_i$;

```
Take out the number of rows of the improvement matrix rows;
for i=1:rows do
    Extract the i-th corresponding data;
    The data, neighborhood radius(ϵ = 10^{-12}) and minimum points (MinPts=10) are used by DBSCAN to generate clustering;
    Record the clustering results and the number of clusters K;
    for m=1:K do
        The distance between data points in each cluster is calculated, and the average distance is used as the density measure;
        The group with less than the average density and more than 100 elements is stored in index;
    end for
    for n=1:index do
        The number of clusters K ∈ [10, 20] is generated randomly;
        Call the k-means clustering function to cluster the low-density group;
    end for
    The clustering results of k-means and DBSCAN were merged to get the grouping result group_i and the average improvement of each group improve_i is recorded;
end for
```

Using DBSCAN first and then k-means for clustering grouping may be more suitable for identifying clusters with non-convex shapes and noise because DBSCAN can handle these situations better. While using k-means first and then using DBSCAN for clustering grouping may be more suitable for identifying dense convex clusters. Therefore, according to specific data characteristics and clustering purposes, the combination of the two can complement each other, thus improving clustering effect and obtaining better clustering results.

The advantage of KDB clustering method and its variant is that, it makes full use of k-means' ability to divide convex clusters and DBSCAN's ability to deal with non-convex clusters and noise, and it can fuse density information and distance information, and can process clusters of various shapes and densities to obtain more comprehensive and accurate clustering results with better robustness and adaptability.

Besides KDB clustering method, we also designed a correlation analysis based decomposition method and a random grouping method. In correlation analysis based decomposition method the Spearman correlation coefficient is used to decompose the large-scale problem. The random grouping method involves partitioning the problem into subproblems, each containing a maximum of 100 variables. The reason why we design four different decomposition methods is that, for fully non-separable large-scale problems, we aim to maximize the diversity of the decomposition to make up for the 'hard decomposition'. The term 'hard decomposition' refers to situations where large-scale problems cannot be divided using traditional techniques due to the complex interactions among variables. To make the optimization more effective and efficient, we have to apply the decomposition strategy to decompose these large-scale problems. Thus, having more decomposition strategies can lead to better decomposition diversity and result in better optimization results.

B. The overall framework for optimizing fully non-separable large-scale problems

The overall algorithm can be described in the following steps. Firstly, we use self-adaptive discrete scan method [30] as a line search method and execute it for 10 times and we can obtain the current best member and best value of the optimization. Also, we can get the improvement matrix of each dimension. Secondly, we use KDB clustering method and its variant to decompose the fully non-separable problem. Thirdly, we use correlation analysis based decomposition method and random grouping method to decompose the fully non-separable problem. Lastly, for all decomposition results, we optimised each subpopulation iteratively using the adaptive DE algorithm SaNSDE [40] and the MATLAB fmincon method until the termination condition was met.

The overall framework can be summarized in Algorithm 3.

IV. Experimental setup and results

In this section, numerical experiments are conducted to test the performance and efficiency of the proposed decomposition method for fully non-separable problems. The experiments are carried out on 6 fully non-separable large-scale problems taken from CEC'2010 and CEC'2013 benchmark suite. These benchmark problems are designed to simulate real-world scenarios and are commonly used to compare algorithm

Algorithm 3 The overall framework

use self-adaptive discrete scan method [30] as a line search method and run it for 10 iterations;
record the best value, best member, the improvement matrix.
while FEs<maxFE **do**
    use the four decomposition strategies proposed to divide the problem and record the results in $group$;
    sort $group$ in descending order;
    for all $group$ **do**
        use SaNSDE [40] to optimize this group;
        use the matlab fmincon to optimize this group;
    end for
end while

performance in LSGO algorithms. Note that, in this paper we focus on fully non-separable large-scale problems, so only these kinds of test problems are chosen, that is, f19 and f20 of CEC'2010 and f12- f15 of CEC'2013 benchmark suite. Among the test problems, f19 and f20 are 1000-dimensional, f13 and f14 are 905-dimensional, f12 and f15 are 1000-dimensional. All the test problems have the global best function value 0. The maximum number of function evaluations (FEs) is fixed at $3e+6$ as required.

To test the efficiency of the proposed decomposition method, we compare its results with that of correct decomposition which is non-decomposition because all variables of fully non-separable problems are interacted with each other and theoretically should not be decomposed. The algorithms were executed 15 times for each test problem using the proposed decomposition method and the non-decomposition. The best function value (Best), median function value (Median), worst function value (Worst), mean function value (Mean), and standard deviation (Std) were recorded for each of these 15 runs. To assess the significance of the results, paired-sample t-test is conducted at the significance level of $\alpha = 0.05$ for all comparisons and the resulting p-values are recorded and the significantly better results are marked in bold. The comparative results are presented in Table I and Table II, in which non-decomposition stands for the correct decomposition and ours stands for the decomposition method proposed in this paper.

TABLE I
EXPERIMENTAL RESULTS ON FULLY NON-SEPARABLE PROBLEMS OF CEC'2010 BENCHMARK SUITE

		non-decomposition	ours	p-value
f19	Best	9.90e+06	**3.12e+03**	1.54e-63
	Median	1.11e+07	**6.70e+03**	
	Worst	1.11e+07	**2.18e+04**	
	Mean	1.08e+07	**8.29e+03**	
	Std	4.96e+05	**4.80e+03**	
f20	Best	4.40e+02	**1.33e-07**	3.17e-36
	Median	7.13e+02	**1.36e-07**	
	Worst	7.13e+02	**1.51e-07**	
	Mean	6.48e+02	**1.37e-07**	
	Std	1.19e+02	**4.01e-09**	

Table I shows the experimental results of non-decomposition and the proposed decomposition method for the two test problems of CEC'2010 benchmark suite. From Table I we can see that the proposed decomposition method performs significantly better than that of non-decomposition. For test function f19, the mean value of the proposed decomposition method can achieve the order of 10^3 magnitude while that of non-decomposition is in the order of 10^7. The proposed method is 4 orders of magnitude better. For test problem f20, the proposed method successfully finds the global optimal solution (with an accuracy error 10^{-7}) while that of non-decomposition only can get a mean value of $6.48e+02$. The proposed decomposition method performs much better than non-decomposition with 9 orders of magnitude. Furthermore, the standard deviation 10^{-9} of the proposed decomposition method indicates a high level of stability of the method. So we can see that the proposed decomposition method is effective and performs and yields superior results for fully non-separable test problems of the CEC'2010 benchmark suite.

TABLE II
EXPERIMENTAL RESULTS ON FULLY NON-SEPARABLE PROBLEMS OF CEC'2013 BENCHMARK SUITE

		non-decomposition	ours	p-value
f12	Best	4.22e+02	**1.37e-07**	2.76e-17
	Median	4.54e+02	**1.39e-07**	
	Worst	9.32e+02	**1.46e-07**	
	Mean	5.75e+02	**1.40e-07**	
	Std	2.12e+02	**2.11e-09**	
f13	Best	5.02e+09	**1.16e+07**	6.84e-04
	Median	9.27e+09	**3.31e+07**	
	Worst	5.33e+10	**3.87e+08**	
	Mean	1.91e+10	**7.82e+07**	
	Std	1.94e+10	**1.03e+08**	
f14	Best	4.41e+10	**2.69e+07**	9.29e-08
	Median	1.54e+11	**4.67e+07**	
	Worst	2.53e+11	**6.16e+07**	
	Mean	1.33e+11	**4.45e+07**	
	Std	7.63e+10	**1.18e+07**	
f15	Best	1.26e+09	**1.55e+07**	1.57e-04
	Median	1.13e+10	**2.24e+07**	
	Worst	6.72e+10	**3.07e+08**	
	Mean	1.75e+10	**8.28e+07**	
	Std	2.07e+10	**8.69e+07**	

Table II shows the comparison results of the fully non-separable problems of CEC'2013 benchmark suite. We can easily see that for all the 4 test problems, the proposed decomposition method performs significantly better than non-decomposition. Especially for f12, the proposed decomposition method gain a near-optimal mean value $1.4e-07$ which is 9 orders of magnitude better than non-decomposition. For test problems f13, the proposed decomposition method is 3 orders of magnitude better, for f14 4 orders and for f15 3 orders better than non-decomposition. So we can say that the proposed decomposition method totally outperformed non-decomposition.

Overall, the proposed decomposition method yielded superior and more stable results than non-decomposition in all the six test problems. Since the only difference of these two approaches is the decomposition strategy: non-decomposition presents the correct decomposition while the proposed decomposition method designed multiple strategy and divide the fully non-separable problem.

To better show the performance of the proposed method, comparisons with four well known and state-of-the-art LSGO algorithms. DECC_DG [2] and CCVIL [24] are well known LSGO algorithms and often used for comparison, while R-LLPSO [35] and AGLDPSO [36] are state-of-the-art LSGO algorithms. The results of fully non-separable problems in the CEC'2010 (f19 and f20) and CEC'2013 (f12-f15) benchmark suites are presented in Table III.

TABLE III
EXPERIMENTAL RESULTS ON FULLY NON-SEPARABLE PROBLEMS OF CEC'2010 AND CEC'2013 BENCHMARK SUITE

		ours	DECC_DG	CCVIL	RLLPSO	AGLDPSO
f12	Mean	**1.40e-07**	8.76e+10	3.90e+07	1.87e+03	1.53e+03
	Std	2.11e-09	1.04e+10	3.28e+07	1.72e+02	2.79e+02
f13	Mean	**7.82e+07**	2.38e+10	8.57e+09	2.17e+08	2.12e+10
	Std	1.03e+08	5.72e+09	3.15e+09	5.46e+07	3.49e+09
f14	Mean	**4.45e+07**	1.16e+11	5.94e+10	**5.06e+07**	9.46e+07
	Std	1.18e+07	3.49e+10	4.05e+10	2.16e+07	5.68e+06
f15	Mean	8.28e+07	1.37e+07	6.40e+06	**1.63e+06**	2.73e+06
	Std	8.69e+07	2.11e+06	1.37e+06	5.88e+04	1.19e+05
f19	Mean	**8.29e+03**	2.17e+06	3.52e+05	8.71e+05	8.43e+05
	Std	4.80e+03	1.22e+05	2.04e+04	2.31e+04	4.61e+04
f20	Mean	**1.37e-07**	9.18e+10	1.11e+03	1.84e+03	1.44e+03
	Std	4.01e-09	8.99e+09	3.04e+02	1.68e+02	1.56e+02

It is evident from the table that the proposed method outperforms the other algorithms, achieving the best results in 5 out of 6 test functions (f12, f13, f14, f19 and f20). RLLPSO attains the best results in two functions (f14, f15), while the remaining algorithms do not achieve any best results. Notably, the proposed method outperforms other algorithms by at least one order of magnitude for test problem f13. For test problem f19, the proposed algorithm achieves the mean value of 10^3 order, which is two-order-of-magnitude better than other compared algorithms. For test problem f14, although the proposed algorithm achieved the best result at the same order-of-magnitude with RLLPSO, it exhibits lower variance, which can lead to more stable optimal results. Moreover, the proposed method can successfully find the global optimal solution for test functions f12 and f20 with an accuracy error around 10^{-7} while no compared algorithms can achieve this goal. Only for f15, the proposed method performs worse than other compared algorithms. So we can get the conclusion that the proposed method significantly outperforming the other four algorithms. These comparison results further demonstrate the effectiveness of the proposed decomposition-based optimization approach for fully non-separable large-scale problems.

The comparative analysis yields a significant finding: while conventional wisdom suggests that decomposition should align with variable correlations or interactions, this approach proves ineffective when applied to fully non-separable large-scale problems. These problems, characterized by intricate variable interdependencies, present a formidable challenge in the realm of optimization and remain unresolved. A reasonable decomposition may shed some light on solving them.

V. CONCLUSIONS

In this paper, a novel decomposition method designed for fully non-separable problems is introduced. To make the decomposition more effective, 4 strategies are designed which can generate more than 20 different results. Experiment are conducted on fully non-separable problems taken from CEC'2010 and CEC'2013 benchmark suite. The experimental results demonstrate that the proposed decomposition method can effectively optimize fully non-separable problems, getting better optimization results. This indicates that, although theoretically decomposition should not be applied to fully non-separable problems, optimization algorithms can get better performance with reasonable decomposition.

ACKNOWLEDGMENTS

National Natural Science Foundation of China (No. 62002289).

REFERENCES

[1] M. A. Potter, K. A. D. Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, Evolutionary Computation 8 (2000) 1–29.

[2] M. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative co-evolution with differential grouping for large scale optimization, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 378–393. doi:10.1109/TEVC.2013.2281543.

[3] Y. Mei, M. N. OMIDVAR, X. Li, X. Yao, Competitive divide-and-conquer algorithm for unconstrained large scale black-box optimization, ACM Transactions on Mathematical Software 42 (2) (2015).

[4] M. N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, Dg2: A faster and more accurate differential grouping for large-scale black-box optimization, IEEE Transactions on Evolutionary Computation PP (99) (2017) 1–1. doi:10.1109/TEVC.2017.2694221.

[5] X.-M. Hu, F.-L. He, W.-N. Chen, J. Zhang, Cooperation co-evolution with fast interdependency identification for large s-cale optimization, Information Sciences 381 (2017) 142 – 160. doi:https://doi.org/10.1016/j.ins.2016.11.013.

[6] M. Yang, A. Zhou, C. Li, X. Yao, An efficient recursive differential grouping for large-scale continuous problems, IEEE Transactions on Evolutionary Computation 25 (1) (2020) 159–171.

[7] X. Ma, Z. Huang, X. Li, L. Wang, Y. Qi, Z. Zhu, Merged differential grouping for large-scale global optimization, IEEE Transactions on Evolutionary Computation 26 (6) (2022) 1439–1451. doi:10.1109/TEVC.2022.3144684.

[8] M. N. Omidvar, B. Kazimipour, X. Li, X. Yao, Cbcc3 - a contribution-based cooperative co-evolutionary algorithm with improved exploration/exploitation balance, in: Proc. 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3541–3548. doi:10.1109/CEC.2016.7744238.

[9] Y. Mei, X. Li, X. Yao, Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 435–449. doi:10.1109/TEVC.2013.2281503.

[10] X. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Transactions on Evolutionary Computation 16 (2) (2012) 210–224. doi:10.1109/TEVC.2011.2112662.

[11] Q. Yang, W. Chen, T. Gu, H. Zhang, J. D. Deng, Y. Li, J. Zhang, Segment-based predominant learning swarm optimizer for large-scale optimization, IEEE Transactions on Cybernetics 47 (9) (2017) 2896–2910. doi:10.1109/TCYB.2016.2616170.

[12] K. M. Mishra, M. Gallagher, A modified screening estimation of distribution algorithm for large-scale continuous optimization, in: G. Dick, W. N. Browne, P. Whigham, M. Zhang, L. T. Bui, H. Ishibuchi, Y. Jin, X. Li, Y. Shi, P. Singh, K. C. Tan, K. Tang (Eds.), Simulated Evolution and Learning, Springer International Publishing, Cham, 2014, pp. 119–130.

[13] W. Dong, T. Chen, P. Tiňo, X. Yao, Scaling up estimation of distribution algorithms for continuous optimization, IEEE Transactions on Evolutionary Computation 17 (6) (2013) 797–822. doi:10.1109/TEVC.2013.2247404.

[14] C. W. Ahn, J. An, J.-C. Yoo, Estimation of particle swarm distribution algorithms: Combining the benefits of pso and edas, Information Sciences 192 (2012) 109 – 119, swarm Intelligence and Its Applications. doi:https://doi.org/10.1016/j.ins.2010.07.014.

[15] S. I. Valdez, A. HernáNdez, S. Botello, A boltzmann based estimation of distribution algorithm, Inf. Sci. 236 (2013) 126–137. doi:10.1016/j.ins.2013.02.040.

[16] D. Molina, M. Lozano, F. Herrera, Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization, in: Proc. IEEE Congr. Evolutionary Computation (CEC), 2010, pp. 1–8. doi:10.1109/CEC.2010.5586034.

[17] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm and Evolutionary Computation 2 (2012) 1–14.

[18] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, M.-H. Lim, Ockham's razor in memetic computing: three stage optimal memetic exploration, Information Sciences 188 (2012) 17–43.

[19] A. LaTorre, S. Muelas, J.-M. Pena, Large scale global optimization: Experimental results with mos-based hybrid algorithms, in: Proc. IEEE Congr. Evolutionary Computation (CEC), 2013, pp. 2742–2749. doi:10.1109/CEC.2013.6557901.

[20] L.-Y. Tseng, C. Chen, Multiple trajectory search for large scale global optimization, in: Proc. IEEE Congr. Evolutionary Computation, 2008, pp. 3052–3059. doi:10.1109/CEC.2008.4631210.

[21] M. Omidvar, X. Li, X. Yao, Cooperative co-evolution with delta grouping for large scale non-separable function optimization, in: Proc. IEEE Congr. Evolutionary Computation (CEC), 2010, pp. 1–8. doi:10.1109/CEC.2010.5585979.

[22] J. Sun, H. Dong, Cooperative co-evolution with correlation identification grouping for large scale function optimization, in: Proc. 2013 International Conference on Information Science and Technology (ICIST), 2013, pp. 889–893. doi:10.1109/ICIST.2013.6747683.

[23] K. Weicker, N. Weicker, On the improvement of coevolutionary optimizers by learning variable interdependencies, in: Proc. IEEE Congr. Evolutionary Computation, Vol. 3, 1999, pp. 1627–1632. doi:10.1109/CEC.1999.785469.

[24] W. Chen, T. Weise, Z. Yang, K. Tang, Large-scale global optimization using cooperative coevolution with variable interaction learning, in: Parallel Problem Solving from Nature, PPSN XI, Vol. 6239 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 300–309. doi:10.1007/978-3-642-15871-1-31.

[25] Y. Sun, M. Kirley, S. K. Halgamuge, A recursive decomposition method for large scale continuous optimization, IEEE Transactions on Evolutionary Computation (2017). doi:10.1109/TEVC.2017.2778089.

[26] J. Fan, J. Wang, M. Han, Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods, IEEE Transactions on Fuzzy Systems 22 (4) (2014) 829–839. doi:10.1109/TFUZZ.2013.2276863.

[27] S. Mahdavi, S. Rahnamayan, M. E. Shiri, Multilevel framework for large-scale global optimization, Soft Computing 21 (14) (2016) 1–30. doi:10.1007/s00500-016-2060-y.

[28] L. Ma, S. Cheng, Y. Shi, Enhancing learning efficiency of brain storm optimization via orthogonal learning design, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (11) (2021) 6723–6742. doi:10.1109/TSMC.2020.2963943.

[29] Y. Wang, H. Liu, F. Wei, T. Zong, X. Li, Cooperative coevolution with formula-based variable grouping for large-scale global optimization, Evolutionary computation 26 (4) (2018) 569–596.

[30] H. Liu, Y. Wang, L. Liu, X. Li, A two phase hybrid algorithm with a new decomposition method for large scale optimization, Integrated Computer-Aided Engineering 25 (4) (2018) 349–367.

[31] H. Liu, Y. Wang, N. Fan, A hybrid deep grouping algorithm for large scale global optimization, IEEE Transactions on Evolutionary Computation 24 (6) (2020) 1112–1124.

[32] H. Liu, Y. Cheng, S. Xue, S. Tuo, A space-reduction based three-phase approach for large-scale optimization, Applied Soft Computing (2023) 110517.

[33] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transactions on Cybernetics 45 (2) (2015) 191–204. doi:10.1109/TCYB.2014.2322602.

[34] Q. Yang, W. Chen, J. D. Deng, Y. Li, T. Gu, J. Zhang, A level-based learning swarm optimizer for large-scale optimization, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 578–594. doi:10.1109/TEVC.2017.2743016.

[35] F. Wang, X. Wang, S. Sun, A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization, Information Sciences 602 (2022) 298–312.

[36] H. Deng, L. Peng, H. Zhang, B. Yang, Z. Chen, Ranking-based biased learning swarm optimizer for large-scale optimization, Information Sciences 493 (2019) 120–137.

[37] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H. Jin, J. Zhang, Adaptive granularity learning distributed particle swarm optimization for large-scale optimization, IEEE Transactions on Cybernetics 51 (3) (2021) 1175–1188.

[38] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, Oakland, CA, USA, 1967, pp. 281–297.

[39] A. Auger, D. Brockhoff, N. Hansen, Investigating the impact of sequential selection in the (1, 2)-cma-es on the noisy BBOB-2010 testbed, in: Proc. Genetic and Evolutionary Computation Conference, 2010, pp. 1605–1610. doi:10.1145/1830761.1830779.

[40] Z. Yang, K. Tang, X. Yao, Self-adaptive differential evolution with neighborhood search, in: Proc. IEEE Congr. Evolutionary Computation, 2008, pp. 1110–1116. doi:10.1109/CEC.2008.4630935.