Cooperative co-evolutionary explicit averaging strategy Differential Evolution with Linkage Measurement Minimization based on LINC-R in Noisy Environment

Rui Zhong
Graduate School of Information Science and Technology
Hokkaido University
Sapporo, Japan
rui.zhong.u5@elms.hokudai.ac.jp

Masaharu Munetomo
Information Initiative Center
Hokkaido University
Sapporo, Japan
munetomo@iic.hokudai.ac.jp

***Abstract*—Many optimization problems suffer from noise, and the noise combined with the large-scale attribution makes the problem complexity explode. Cooperative co-evolution (CC) based on divide and conquer decomposes the problems and solves the sub-problems alternately, which is a popular framework for solving large-scale optimization problems. Meanwhile, many studies show that the CC framework is sensitive to decomposition. However, the high accuracy decomposition methods based on perturbation such as Linkage Identification by nonlinearity check on real code (LINC-R), Differential grouping (DG) are sensitive to sampling accuracy and will fail to detect the interactions in noisy environments. Therefore, solving large-scale optimization problems in noisy environments based on the CC framework faces unprecedented challenges. In this paper, we propose a novel method named Linkage Measurement Minimization based on LINC-R by Elitist Genetic Algorithm (EGALINC-Rmin). In the decomposition stage, we regard the decomposition problem as an optimization problem and design the linkage measure function based on LINC-R. Since the linkage measure function is multimodal, we attach a penalty to lead the direction of optimization. In the optimization stage, we hypothesize that the elite sub-population plays a decisive role in the optimization, and we introduce a fitness-based explicit averaging strategy with dynamic sampling to correct the linkage measure value of the elite sub-population in noisy environments. Numerical experiments show that our proposal is competitive compared with existing decomposition methods in noisy environments, and the introduction of an explicit averaging strategy (EAS) can enhance the anti-noise ability of the optimizer.**

***Index Terms*—Cooperative Coevolution (CC), Linkage Measurement Minimization, explicit averaging strategy (EAS), Large-scale optimization, noisy environments**

I. Introduction

Evolutionary Algorithms (EAs) based on Darwin's theory of evolution are employed to solve complex optimization problems with great success [1], [2]. However, noise widely exists in the fitness evaluation of many problems [3]–[5], which often makes the difference between the fitness evaluation observation and the real fitness observation. Meanwhile, noise can mislead the direction of optimization, which makes it difficult to be solved with traditional EAs. In the past decade, many studies [6], [7] on optimization problems in noisy environments have been published, and some strategies have been introduced to traditional EAs to tackle the noise. Examples include explicit averaging [8], implicit averaging [9], Fourier transform [10], fitness estimation [11], and more. Most of the previous research focuses on relatively low-dimensional problems (up to 100-D), and few studies on noisy problems with large-scale optimization problems have been published. In fact, many noisy optimization problems are high-dimensional, such as parameters and structures optimization of Deep Neural Networks [12] and subset selection [13].

The large-scale optimization problem in noisy environments contains challenges both on the large scale and noise, which make the difficulties of the problem-solving explosive. The main reasons are the following aspects: (1) The complexity of the optimization problem increases, including the increase of dimensionality and the existence of the noise. (2) The search space of large-scale problems increases exponentially with the increase of dimensionality, which is known as the curse of dimensionality [14]. (3) The computational cost of building a surrogate model is expensive, and the accuracy is also affected by noise and the curse of dimensionality, which makes some algorithms limited [15], [16].

Many algorithms have been proposed to overcome the challenge of the large-scale optimization problems, such as designing optimization operators to adapt the large-scale attributes [17], building surrogate models [18], and decomposing the problem [19], which is known as the cooperative coevolution (CC). In this paper, we apply the CC framework to solve large-scale optimization problems in noisy environments. This method is inspired by the divide and conquer, which has achieved great success in solving large-scale continuous [20], combinatorial [21], and constrained [22] problems.

How to group the variables of large-scale problems is the key to the CC framework. Many studies [23], [24] show that the CC framework is sensitive to problem decomposition strategies. Currently, the decomposition strategies developed based on perturbations such as Linkage Identification by Nonlinearity Check (LINC-R) and Diffential Grouping (DG) are considered high accuracy. These clustering methods perturb candidate variables with δ and detect the interaction by fitness difference. However, a large number of fitness evaluation consumptions and local linearity checks under the background of large scale make the scalability of these algorithms limited. More seriously, these variable grouping algorithms are extremely sensitive to sampling accuracy on the fitness landscape, and interactions between variables will completely fail to be detected in multiplicative noise environments. We will explain the reason in Section II.

In this paper, we propose a novel grouping strategy named Linkage Measurement Minimization based on LINC-R by Elitist Genetic Algorithm (LINC-Rmin), our proposal allows an automatic decomposition that treats the variable grouping problem as a combinatorial optimization problem and we design an linkage measure function based on LINC-R. According to our previous research, we found that the linkage measure function is multimodal, so we attach a penalty to lead the direction of optimization. In addition, we introduce a fitness-based explicit averaging strategy (EAS) with dynamic sampling to the traditional Differential Evolution Algorithm (DE) to enhance the anti-noise ability of the optimizer (EASDECC-LINC-Rmin). More specifically, the main contributions of this paper are as follows.

1. This paper provides a theoretical support. We mathematically explain the feasibility of our design of linkage measure function and the relationship with LINC-R.

2. The mechanism analysis shows how our proposal overcomes the noisy environments, Numerical experiments also show that our proposal can solve large-scale optimization problems in noisy environments better than some state-of-the-art variable grouping methods.

3. Our proposal provides a novel strategy for solving the variable grouping problem, which can be extended for higher-dimensional, multi-objective, real-world problems with simple adjustments.

4. We introduce an efficient anti-noise strategy into the optimizer. Experiments show that our strategy can enhance the anti-noise ability of the optimizer to a certain extent. Theoretical analysis proves that our strategy is effective and can be easily extended to almost all optimizers.

The rest of the paper is organized as follows, Section II covers preliminaries and a brief review both on variable grouping methods and EAS, and reveals the challenges of perturbation-based grouping methods in multiplicative noise environments. Section III provides a detailed introduction of our proposal. Section VI describes the experiments on CEC2013 LSGO Suite in noisy environments and analyzes the experimental results. Section V concludes the paper and shows future directions.

II. Preliminaries and related work

A. Preliminaries

1) Large-scale Optimization Problem: Without loss of generality, a large-scale optimization problem can be defined as follows:

$$\min f(\mathrm{X})$$
$$s.t. : \mathrm{X} \in \mathbb{R}$$

Where $\mathrm{X} = (x_1, x_2, ..., x_n)$ is an n-dimensional decision vector, and each x_i $(i \in [1, n])$ is a decision variable. $f(\mathrm{X})$ is the objective function needed to be minimized. In our work, the large-scale optimization problem is a special case of black-box optimization, where the number of decision variables n is large (e.g., $n \geqslant 1000$).

2) Variable Interaction: The concept of variable interaction is derived from biology. In biology, if a feature at the phenotype level is contributed by two or more genes, then we consider there are interactions between these genes, and the genome composed of these genes is called a linkage set [25]. In the definition of optimization problems. If $\min\ (f(x_1, x_2, ..., x_n)) = \min\ (\sum_{i=1}^{m} f(x_{i_1}, ..., x_{i_k})$, then $f(x)$ is a partially separable function, and $L_i = [x_{i_1}, ..., x_{i_k}]$ is called linkage set. There are two extreme cases, when there is no interaction between all variables, which means $\min\ (f(x_1, x_2, ..., x_n)) = \min\ (\sum_{i=1}^{n} f(x_{i_1})$, then we call $f(x)$ is a fully separable function. On the contrary, we call $f(x)$ is a completely nonseparable function if all variables have direct or indirect interactions with each other.

3) Noise in environments: Additive noise [26] and multiplicative noise [27] widely exist in the evaluation of optimization problems. Mathematically, the noisy objective function $f^N(\mathrm{X})$ of a trial solution X is represented by

$$f^N(\mathrm{X}) = f(\mathrm{X}) + \eta \tag{1}$$

$$f^N(\mathrm{X}) = f(\mathrm{X}) \cdot (1 + \beta) \tag{2}$$

where $f(\mathrm{X})$ is the real objective function. Eq (1) shows the objective function in addictive noisy environments, η is the amplitude of the addictive noise. Eq (2) reveals the relationship between the real objective function and objective function in multiplicative noisy environments. β is a random noise (such as Gaussian noise).

4) Differential Evolution Algorithm: The Differential Evolution Algorithm (DE) was first proposed by Storn and Price in 1995 [28]. DE has been applied to solve many complex optimization problems. Similar to the Genetic Algorithm, DE mainly includes mutation, crossover, and selection. The Pseudocode of DE is shown in Algorithm 1

Algorithm 1: Pseudocode of DE Algorithm

Input: Population : P; Dimension : D; Generation : T; Prior knowledge : K
Output: Offspring : OP

Function **DE** (P, D, T, K) **:**
  $t \leftarrow 0$
  if K *is not None* **then**
    $P \leftarrow$ **Update**(P, K)
  end
  while not stop criterion **do**
    for $i = 0$ *to* M **do**
      ▶ (Mutation and crossover)
      for $j = 0$ *to* D **do**
        $v_{i,t}^{j} \leftarrow$ **Mutation**$(P_{i,t}^{j})$
        $u_{i,t}^{j} \leftarrow$ **Crossover**$(P_{i,t}^{j}, v_{i,t}^{j})$
      end
      ▶ (Greedy selection)
      $f_1 \leftarrow$ **Evaluate**$(u_{i,t})$
      $f_2 \leftarrow$ **Evaluate**$(P_{i,t})$
      if $f_1 < f_2$ **then**
        $P_{i,t} \leftarrow u_{i,t}$
      end
      $OP \leftarrow P_{i,t}$
      $t \leftarrow t + 1$
    end
  end
  return OP

In the procedure of DE, mutation is an essential process. We briefly introduce 4 common mutation strategies:

$$\begin{aligned}
\textbf{DE/rand/1}: V_i(g) &= X_{p1}(g) + F \cdot (X_{p2}(g) - X_{p3}(g)) \\
\textbf{DE/best/1}: V_i(g) &= X_{best}(g) + F \cdot (X_{p1}(g) - X_{p2}(g)) \\
\textbf{DE/rand/2}: V_i(g) &= X_{p1}(g) + F \cdot (X_{p2}(g) - X_{p3}(g)) \\
&\quad + F \cdot (X_{p4}(g) - X_{p5}(g)) \\
\textbf{DE/best/2}: V_i(g) &= X_{best}(g) + F \cdot (X_{p1}(g) - X_{p2}(g)) \\
&\quad + F \cdot (X_{p3}(g) - X_{p4}(g))
\end{aligned}$$

$X_{pi}(g)$ is the different individuals randomly selected from the current population, $X_{best}(g)$ is the best individual in the current population. F is the scaling factor. The differential vectors between individuals are calculated in the mutation is the origin of the name of DE.

B. A brief review of the grouping method

Based on the divide and conquer, the CC framework decomposes the large-scale optimization problem into several nonseparable sub-problems and optimizes the sub-problems alternately, which is the mainstream framework for solving large-scale optimization problems. Cooperative Coevolutionary Genetic Algorithm (CCGA) [19], which is proposed with CC framework, applies a simple strategy that decomposes the n-D problem to $n \times 1$-D problems. This method completely ignores the interactions between variables. In theory, it will outperform on fully separable functions compared with standard GA, but will be unacceptable for nonseparable functions. The experimental results show that CCGA performs well on four fully separable functions such as Rastrigin, and obtains the expected results on fully nonseparable functions like Rosenbrock. Random grouping [29] also applies a simple strategy that randomly decomposes an n-D problem into $m \times k$-D sub-problems ($n = m \times k$), where m and k are specified by the user. Mathematically, in multiple trial experiments, the probability that two interacting variables are divided into the same sub-problem at least once is quite high.

With little or no knowledge about the fitness landscape, Delta grouping [30] notices that the difference in coordinates from the initial population to the optimized population is larger when x_i and x_j are separable variables, and the difference is smaller when x_i and x_j are nonseparable. Based on this phenomenon, delta grouping optimizes the random population for one generation and calculates the difference of each dimension according to Eq (3).

$$i \in [1, n], \bar{\delta_i} = \frac{\sum_{j=1}^{N_{pop}} \delta_{i,j}}{N_{pop}} \tag{3}$$

$\delta_{i,j}$ denotes the difference of the j^{th} individual on the i^{th} dimension, N_{pop} is the population size. DECC-D decomposes the problem into $m \times k$-D subproblems ($n = m \times k$) according to the sorted dimensional differences. The experimental results on CEC2008 and CEC2010 Suites show that this rough estimation can capture interactions with little prior knowledge.

As the pioneers of perturbation-based grouping methods, Linkage Identification by non-Monotonicity Detection (LIMD) [31] and Linkage Identification by Nonlinearity Check (LINC-R) [25] were proposed in 1999 and 2004. Eq (4) defines the perturbation in the i^{th} dimension and the j^{th} dimension.

$$\begin{aligned}
s &= (x_1, x_2, ..., x_n) \\
s_i &= (x_1, ..., x_i + \delta, ..., x_n) \\
s_j &= (x_1, ..., x_j + \delta, ..., x_n) \\
s_{ij} &= (x_1, ..., x_i + \delta, ..., x_j + \delta, ..., x_n)
\end{aligned} \tag{4}$$

LIMD checks the monotonicity based on the random population. Eq (5) defines the principle of LIMD

$$\begin{aligned}
&\exists s \in Pop: \\
&if\ \neg(f(s) < f(s_i) < f(s_{ij})\ and\ f(s) < f(s_j) < f(s_{ij}) \\
&or\ f(s) > f(s_i) > f(s_{ij})\ and\ f(s) > f(s_j) > f(s_{ij})) \\
&then\ x_i\ and\ x_j\ are\ nonseparable
\end{aligned} \tag{5}$$

LIMD checks the monotonicity based on the random population. When x_i and x_j satisfy the simultaneous increase or decrease at all individuals, LIMD identifies x_i and x_j as separable variables.

Meanwhile, LINC-R calculates the fitness difference of perturbation to identify the interaction between variables. More specifically,

$$\begin{aligned}&\exists s\in Pop:\\ &\Delta f_i=f(s_i)-f(s)\\ &\Delta f_j=f(s_j)-f(s)\\ &\Delta f_{ij}=f(s_{ij})-f(s)\\ &if|\Delta f_{ij}-(\Delta f_i+\Delta f_j)|>\varepsilon\\ &then\ x_i\ and\ x_j\ are\ nonseparable\end{aligned}\tag{6}$$

ε is the allowable error. LINC-R is applied to the low-dimensional test function. Based on the ideas of LINC-R, Differential grouping (DG) extends Eq (6) firstly on large-scale optimization problems up to 1000-D. At present, the Linc-R/DG-based grouping methods are considered high accuracy, graphDG [32] even achieved 100% accuracy on most test functions in the CEC2010 LSGO Suite.

C. Challenges of Grouping Methods in Noisy Environments

Additive noise and multiplicative noise are two representative noises. Additive noise is often irrelevant to the fitness landscape, so we can carefully adjust the parameters to overcome the additive noise in decomposition stage, although it is not easy [33]. However, multiplicative noise is related to the fitness landscape, so fitness can amplify the noise. Dealing with multiplicative noise is more difficult than additive noise in decomposition stage. Taking the LINC-R as an example.

$$\begin{aligned}&\exists s\in Pop:\\ &\Delta_1=f^N(s_i)-f^N(s)=f(s_i)(1+\beta_1)-f(s)(1+\beta_2)\\ &\Delta_2=f^N(s_{ij})-f^N(s_j)=f(s_{ij})(1+\beta_3)-f(s_j)(1+\beta_4)\\ &if|\Delta_2-\Delta_1|>\varepsilon\\ &then\ x_i\ and\ x_j\ are\ nonseparable\end{aligned}\tag{7}$$

$\beta_1,\beta_2,\beta_3,\beta_4$ are Gaussian noise. We can see that in noiseless environments, LINC-R can determine the interactions between variables. However, in noisy environments with multiplicative noise, theoretically, the probability of Eq (7) being satisfied is almost 0. In practice, the decomposition methods developed on the LINC-R, such as DG, DG2, RDG, etc. will fail in environments with multiplicative noise. Therefore, grouping methods that detect interactions by perturbation face severe challenges.

D. Explicit averaging strategy for noise

The uncertainty in the estimation of the fitness $f^N(\mathrm{X})$ for an individual X can be reduced by re-evaluating the objective function $f^N(\mathrm{X})$ several times according to Monte Carlo integration [34]. Let the re-evaluating times for $f^N(\mathrm{X})$ be m and $f_i^N(\mathrm{X}))$ represents the ith re-evaluation value. Then we apply the principle of Monte Carlo integration, the mean fitness estimation $f^N(\mathrm{X})$, standard deviation $\sigma(f^N(\mathrm{X}))$ and the standard error of the mean fitness $se(f^N(\mathrm{X}))$ are calculated as

$$\begin{aligned}\bar{f}^N(\mathrm{X})&=\frac{1}{m}\sum_{i=1}^{m}f_i^N(\mathrm{X})\\ \sigma(f^N(\mathrm{X}))&=\sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(f_i^N(\mathrm{X})-\bar{f}^N(\mathrm{X}))}\\ se(\bar{f}^N(\mathrm{X}))&=\frac{\sigma(f^N(\mathrm{X}))}{\sqrt{m}}\end{aligned}\tag{8}$$

It is evident from Eq (8) that sampling an individual ' s objective function m times can reduce $se(\bar{f}^N(\mathrm{X}))$ by a factor of m, thus improving the accuracy in the mean fitness estimation. Roughly, sampling strategies can be primarily classified into two categories based on adaptivity. a). Static sampling: It allocates equal re-evaluation times to all samples. However, the sampling requirements of different trial solutions in different regions in the search space may be different. b). Dynamic sampling: It allocates different re-evaluation times to samples based on their sampling requirements, and the sampling requirements of a candidate solution can be primarily influenced by one or more of the five vital principles including [35], (a) fitness variance, (b) periodicity in updating sample size, (c) number of objectives, (d) characteristics of the optimization problem and (e) comparative analysis with other population members.

III. EASDECC-LINC-RMIN

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

In the flowchart of our proposal, there are two stages including the decomposition stage and the optimization stage. In the decomposition stage, we regard the variable grouping problem as a combinatorial optimization problem and design an linkage measure function based on LINC-R. Then we apply EGA to optimize this problem. In the optimization stage, we alternately apply DE to optimize sub-problems, and after each generation of optimization is finished, we apply a fitness-based explicit averaging strategy to correct the objective value for the elite sub-population. Then we find the best individual from the corrected population and transfer it to the next sub-problem optimization as the prior knowledge.

First, we give a detailed illustration of the decomposition stage. In our previous work, we transform the LINC-R to the additive form of the vector and derived the additive LINC-R to the n-dimensional form. The specific derivation process is as follows.

The original LINC-R is defined as Eq (9)

$$\begin{aligned}&\exists s\in Pop:\\ &if|(f(s_{ij})-f(s_i))-(f(s_j)-f(s))|>\varepsilon\\ &then\ x_i\ and\ x_j\ are\ nonseparable\end{aligned}\tag{9}$$

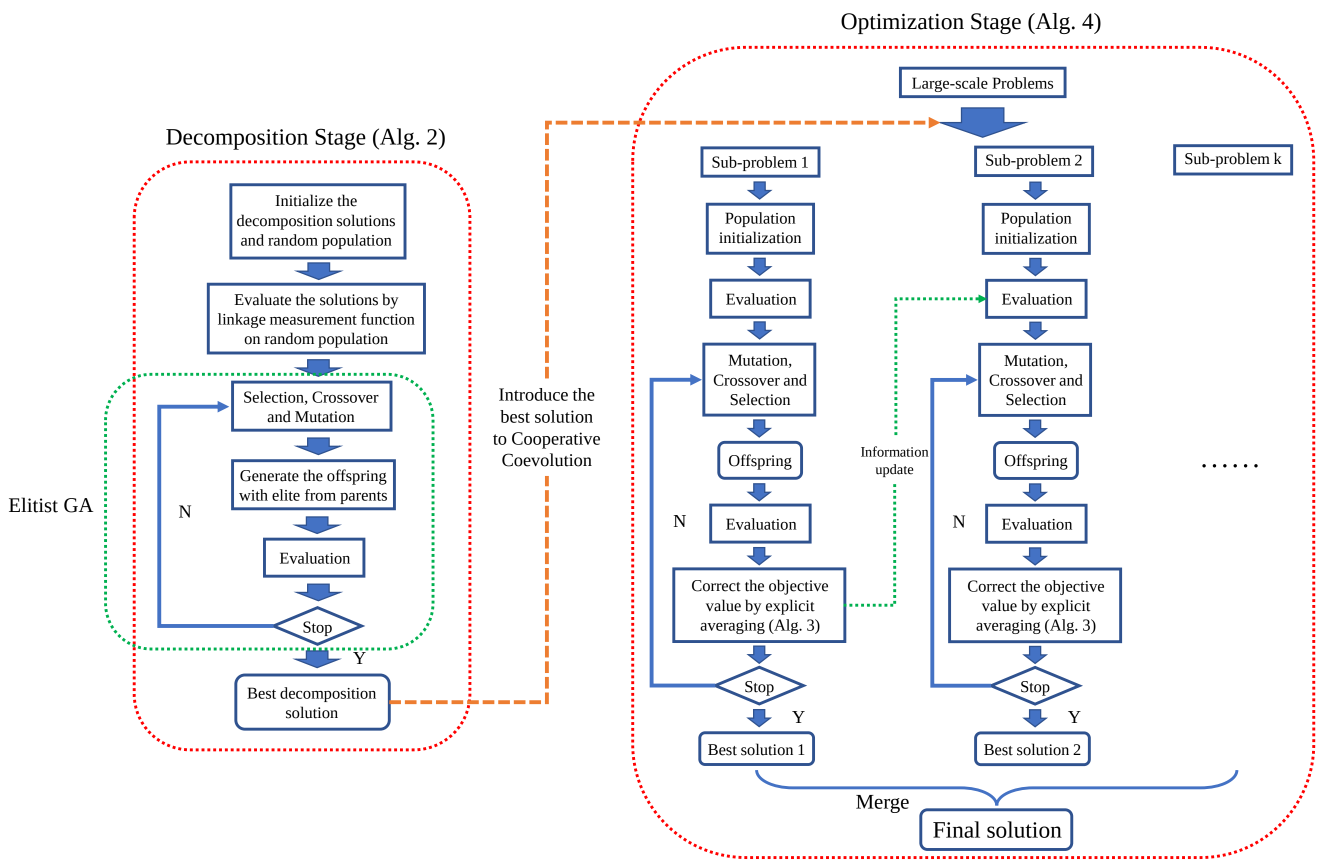


Fig. 1. The flowchart of EASDECC-LINC-Rmin

In our proposal, a variant LINC-R is applied. Eq (10) shows the variant LINC-R

$$
\begin{aligned}
&\exists s \in Pop: \\
&if|(f(s_{ij}) - f(s)) - ((f(s_i) - f(s)) + (f(s_j) - f(s)))| > \varepsilon \\
&then\ x_i\ and\ x_j\ are\ nonseparable
\end{aligned} \tag{10}
$$

Fig. 2 shows the mechanism of the original version of LINC-R and the variant LINC-R work on the separable variables x_i and x_j with one sample s.

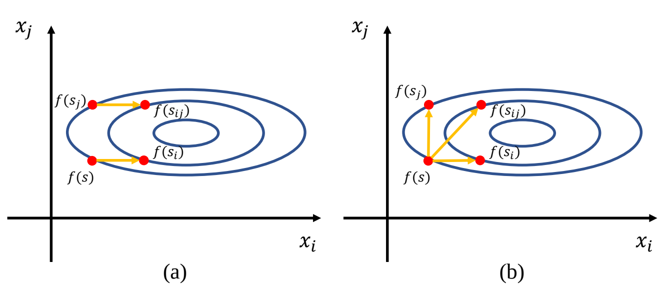


Fig. 2. (a).The mechanism of the original LINC-R works on the separable variables. (b).The mechanism of the variant LINC-R works on the separable variables.

Based on this interesting finding, we next derive LINC-R to 3 or higher dimensions. In 3-dimensional space, the schematic diagram is shown in Fig. 3. Here, we define

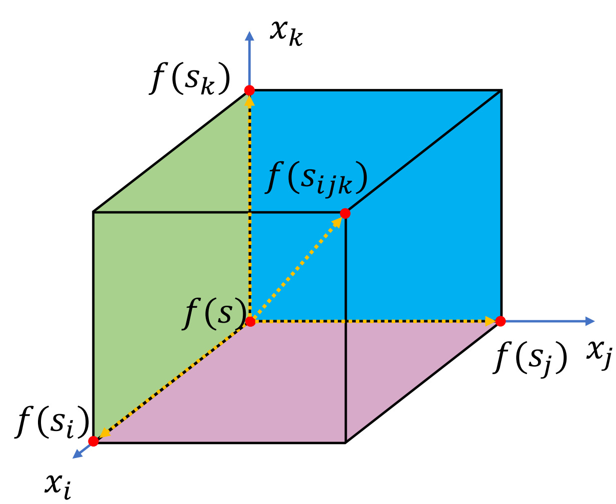


Fig. 3. The variant LINC-R works on 3-D space

$$
\begin{aligned}
\Delta_i &= f(s_i) - f(s) \\
\Delta_j &= f(s_j) - f(s) \\
\Delta_k &= f(s_k) - f(s) \\
\Delta_{ijk} &= f(s_{ijk}) - f(s)
\end{aligned} \tag{11}
$$

When the variant LINC-R is applied simultaneously to determine the interactions between x_i, x_j, and x_k:

$$
\begin{aligned}
&\exists s \in Pop: \\
&if|\Delta_{ijk} - (\Delta_i + \Delta_j + \Delta_k)| > \varepsilon \\
&then\ interaction(s)\ exist\ in\ x_i, x_j, x_k
\end{aligned} \tag{12}
$$

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

$$
\begin{aligned}
&\exists s \in Pop: \\
&if|\Delta_{1,2,...,n} - (\Delta_1 + \Delta_2 + ... + \Delta_n)| > \varepsilon \\
&then\ interaction(s)\ exist\ in\ x_1, x_2, ..., .x_n
\end{aligned} \tag{13}
$$

From the above explanation, we only detect the interaction by LINC-R on $s \in Pop$. For example, if there is no interaction between x_i and x_j, then we say x_i and x_j are separable. Notice that we only check the interactions based on finite samples, although this strategy is limited especially in trap functions, it is impossible to check the interactions based on the whole fitness landscape. And in large-scale problems, it is usual to check the interactions based on one sample due to the limited evaluation times [36], [37]. Thus, Eq (14) is approximately correct in large-scale problems.

$$
\begin{aligned}
&\forall s \in Pop: \\
&if|\Delta_{1,2,...,n} - (\Delta_1 + \Delta_2 + ... + \Delta_n)| < \varepsilon \\
&then\ x_1, x_2, ..., x_n\ are\ fully\ separable
\end{aligned} \tag{14}
$$

However, when Eq (14) is not satisfied, we only know that interactions exist in some variables, but we cannot know in which pairs of variables. Taking 3-D space as an example:

$$
\begin{aligned}
&if\ \exists s \in Pop:\ |\Delta_{ijk} - (\Delta_i + \Delta_j + \Delta_k)| > \varepsilon\ and \\
&\forall s \in Pop:\ |\Delta_{ijk} - (\Delta_{ij} + \Delta_k)| < \varepsilon \\
&then\ x_i, x_j\ are\ nonseparbale \\
&and\ x_k\ is\ separable\ from\ x_i, x_j
\end{aligned}
$$

Therefore, in the n-dimensional space, although it is difficult to detect the interactions between multiple variables through high-dimensional LINC-R directly, we can actively search for the interactions between variables through heuristic algorithms. According to the above description, in the n-dimensional problem, the linkage measure function in our previous proposal [38] is defined in Eq (15)

$$
\min\Big(\sum_{s \in Pop} (\Delta_{1,2,...,n} - \sum_{i,j,...}^{m} (\Delta_{i,j,...}))^2\Big) \tag{15}
$$

m is the number of sub-problems. Comparing with the popular purturbation-based decomposition methods like DG, our proposal allows the linkage measurement minimization with multiple samples in high dimensions, which makes our proposal more robust. However, we found that this linkage measure function is multimodal and usually includes many optima especially in partially separable functions and fully separable functions. Take a simple example. $f(x) = x_1 + (x_2 + 0.5x_3)^2 + 0.5x_4^2$ is a partially separable function, then $\{(x_1), (x_2, x_3), (x_4)\}, \{(x_1, x_2, x_3), (x_4)\}, \{(x_1), (x_2, x_3, x_4)\}$, $\{(x_1, x_4), (x_2, x_3)\}, \{(x_1, x_4, x_2, x_3)\}$are all global optima of this linkage measure function. However, we want the decomposition to be $\{(x_1), (x_2, x_3), (x_4)\}$. We can lead the direction of optimization by a penalty term. We propose Eq (16) as the improved linkage measure function with a penalty.

$$
\begin{aligned}
\min\Big(\frac{\lambda}{|Pop|} \sum_{s \in Pop} |(1 - \sum_{i,j,...}^{m} \frac{\Delta_{i,j,..}}{|\Delta_{1,2,...,n}| + \epsilon})| + \\
\frac{1-\lambda}{num\ of\ group}\Big)
\end{aligned} \tag{16}
$$

$|Pop|$ is the population size, $\frac{1-\lambda}{num\ of\ group}$ is the penalty term, λ is a hyperparameter decided by the user to balance the intensity of penalty. Besides, we normalize the first half of the linkage measure function to eliminate the influence of the dimension on the linkage measure function. We also attach a small value ϵ to the denominator to prevent the denominator from being 0. It can see that when the first half of the linkage measure function is the same, the solution with larger number of groups has better fitness.

In the optimization, Elitist GA is employed to optimize the linkage measure function, and the pseudocode of decomposition stage is shown in Algorithm 2

The elitist strategy was proposed by De Jong [39], which directly replicates the best individual to the next generation without crossover, mutation, and selection. This strategy can prevent the elites from destroying the superior gene and chromosome structure during optimization.

In the optimization stage, we alternately optimize the sub-problems and re-evaluate the elite sub-population to reduce the standard error caused by noise. According to Monte Carlo integration, the standard error in noisy environments can be decreased by the re-evaluation times increase. But under the fitness evaluation times (FEs) limitation, it is wasteful and unnecessary to correct the linkage measure value of all individuals. Based on this hypothesis, we select the best 5% individuals of the population as an elite sub-population in each generation and assign the computational resources according to their fitness based on a roulette strategy. The pseudocode of linkage measure value correction is shown in Algorithm 3. After the linkage measure value correction, we find the best individual in the current population as the prior knowledge for the next sub-problem optimization. The pseudocode of optimization stage is shown in Algorithm 4

IV. Numerical experiment and analysis

In this Section, we ran many experiments to evaluate our proposal, EASDECC-LINC-Rmin. In Section IV-A, we introduce the experiment settings, including benchmark functions, comparing methods, parameters of algorithms, and the performance indicators. Section IV-B introduces the pre-experiments

Algorithm 2: The Pseudocode of Decomposition Stage

Input: Dimension : D; Population size : s_1; Sample size : s_2; Gene length : L; Generation : T
Output: The best decomposition : E

Function `EGALINC-Rmin` (D, s_1, s_2, L, T) **:**
  $t \leftarrow 0$
  ▶ (Initialization)
  for $i = 0$ *to* s_1 **do**
    for $j = 0$ *to* D **do**
      $n \leftarrow$ **randint**$(0, 2^{L-1} - 1)$
      $P_{i,t}^{n} \leftarrow j$
    end
  end
  $S \leftarrow$ **randSamples**(s_2)
  $F_t \leftarrow$ **Evaluate**(P_t^n, S)
  $E \leftarrow$ **bestIndividual**(P_t^n)
  ▶ (Optimization)
  while not stop criterion **do**
    $P_{t+1} \leftarrow$ **Selection**(P_t, F_t, M)
    $P_{t+1} \leftarrow$ **Crossover**(P_{t+1})
    $P_{t+1} \leftarrow$ **Mutation**(P_{t+1})
    $F_{t+1} \leftarrow$ **Evaluate**(P_{t+1}, S)
    $P_{t+1} \leftarrow$ **Replace**(P_{t+1}, E)
    $E \leftarrow$ **bestIndividual**(P_{t+1})
    $t \leftarrow t + 1$
  end
  return E

for determining the intensity of penalty. In Section IV-C, we provide the convergence curve and fitness evaluation consumption in decomposition stage of EASDECC-LINC-Rmin with comparing methods. Finally, we analyze our proposal both in decomposition stage and optimization stage in Section IV-E.

A. Experiment Settings

1) Benchmark Functions: We design 15 test functions in noisy environments based on CEC2013 LSGO Suite, Eq (17) is the definition.

$$f_i^N(x) = f_i(x) \cdot (1 + \beta),\ \ i \in [1, 15] \tag{17}$$

$\beta \sim N(0, 0.01)$. Briefly, this benchmark suite consists 15 test functions with 4 categories.

(1) $f_1^N(x)$ to $f_3^N(x)$: fully separable functions in noisy environments;

(2) $f_4^N(x)$ to $f_7^N(x)$: partially separable functions with 7 none-separable parts in noisy environments;

(3) $f_8^N(x)$ to $f_{11}^N(x)$: partially separable functions with 20 none-separable parts in noisy environments;

(4) $f_{12}^N(x)$ to $f_{15}^N(x)$: functions with overlapping subcomponents in noisy environments;

2) Compareing methods and Parameters: In our experiment design, we compare the decomposition strategy of our proposal with different grouping methods, The algorithms

Algorithm 3: Pseudocode of Objective Value Correction

Input: Computational resource : C; Population : P
Output: Corrected Population : CP

Function `OVC` (C, P) **:**
  $N \leftarrow$ **size**(P)
  ▶ (Sort by fitness in ascend)
  for $i = 0$ *to* N **do**
    for $j = i + 1$ *to* N **do**
      if $P_i.Fit < P_j.Fit$ **then**
        $temp \leftarrow P_i$
        $P_i \leftarrow P_j$
        $P_j \leftarrow temp$
      end
    end
  end
  $EN \leftarrow$ **max**$(1,$ **int**$(N * 0.05))$
  $EP \leftarrow P[0 : EN]$
  $CP \leftarrow P[EN : N]$
  $D \leftarrow 0$
  for $i = 0$ *to* EN **do**
    $D \leftarrow D + P_i.Fit$
  end
  for $i = 0$ *to* EN **do**
    $G_i \leftarrow P_i.Fit / D$
  end
  ▶ (Calculate cumulative probability)
  $p_0 \leftarrow 0$
  for $i = 1$ *to* $EN + 1$ **do**
    $p_i \leftarrow p_{i-1} + G_{i-1}$
  end
  ▶ (Resource allocation by Roulette)
  for $i = 0$ *to* C **do**
    $r \leftarrow$ **rand**$(0, 1)$
    for $j = 1$ *to* $EN + 1$ **do**
      if $p_{j-1} \leqslant r < p_j$ **then**
        $R_{j-1} \leftarrow R_{j-1} + 1$
        Continue
      end
    end
  end
  ▶ (Objective value correction)
  for $i = 0$ *to* EN **do**
    $Obj \leftarrow 0$
    for $j = 0$ *to* R_i **do**
      $Obj \leftarrow Obj +$ **Evaluate**(EP_i)
    end
    $EP_i.Obj \leftarrow (EP_i.Obj + Obj)/(R_i + 1)$
  end
  $CP \leftarrow CP + EP$
  return CP

Algorithm 4: Pseudocode of Optimization Stage

Input: Dimension : D; Gene length : L; Problem : $f(x)$; Computational resources : C; Decomposition population size : M_1; Optimization population size : M_2; Decomposition generation : T_1; Optimization generation : T_2

Output: The best solution : E

Function $\texttt{OPT}(T, OP)$ **:**
- $SP \leftarrow$ **EGALINC-Rmin**$(D, M_1, L, T_1, f(x))$
- $t \leftarrow 0$
- $N \leftarrow$ **size**(SP)
- $K \leftarrow None$ // Prior knowledge
- ▶ (Initialization)
- **for** $i = 0$ *to* N **do**
  - $P_{i,t} \leftarrow$ **popInitial**(SP_i, M_2)
  - $E \leftarrow$ **bestIndividual**$(P_{i,t})$
  - $K \leftarrow$ **Update**(K, E)
- **end**
- ▶ (Optimization)
- **while** $t < T_2$ **do**
  - **for** $i = 0$ *to* N **do**
    - $SD \leftarrow$ **size**(SP_i)
    - $P_{i,t+1} \leftarrow$ **DE**$(P_{i,t}, SD, 1, K, f(x))$
    - $P_{i,t+1} \leftarrow$ **OVC**$(C, P_{i,t+1})$
    - $E \leftarrow$ **bestIndividual**$(P_{i,t+1})$
    - $K \leftarrow$ **Update**(K, E)
  - **end**
  - $t \leftarrow t + 1$
- **end**
- **return** E

applied in the comparisons are listed in Table I, and we also conduct the experiment between EASDECC-LINC-Rmin and DECC-LINC-Rmin to show the effect of the introduction of EAS. Table II shows the parameters of our proposal in the decomposition stage, Table III shows the parameters of the optimization stage.

TABLE I
A SUMMARY OF THE ALGORITHMS UNDER COMPARISON

Algorithms	Optimizer	Decomposition methods
EASDECC-D		Delta grouping [30]
EASDECC-DG		Differential grouping [36]
EASDECC-G	EASDE	Random grouping [29]
EASDECC-VIL		Variable interaction learning [40]
EASDECC-LINC-Rmin		EGALINC-Rmin
DECC-LINC-Rmin	DE	

3) Performance Indicators: We apply the Kruskal-Wallis test to the fitness at the end of the optimization in 25 trial runs between different decomposition methods. If significance exists, then we apply the p-value acquired from the Mann-Whitney U test to do the Holm test. If our proposal is significantly better than the second-best algorithm, we mark

TABLE II
THE PARAMETERS OF DECOMPOSITION OPTIMIZATION

Parameter	value
Optimization direction	Minimization
Optimizer	Elitist GA
Population size	20
Max iteration	20
Gene length	8
λ	0.01
ϵ	10e-9

TABLE III
THE PARAMETERS OF SUB-PROBLEMS OPTIMIZATION

Parameter	value
Optimization direction	Minimization
Optimizer	DE/current-to-best/1
FEs (decomposition+optimization)	3,000,000
Linkage measure value correction cost	3 $\times$ Dimensions
Population size	30 $\times$ Dimensions
Scale factor	0.7
Crossover rate	0.9

* (significance level 5%) or ** (significance level 10%) in the convergence curve. Meanwhile, we apply Mann-Whitney U test between EASDECC-LINC-Rmin and DECC-LINC-Rmin. If EASDECC-LINC-Rmin is significantly better than the method without EAS, we mark * or ** at the end of optimization.

B. Pre-experiment for determining the penalty intensity

In our proposal, we control the intensity of the penalty by λ. Our basic hypothesis is that in fully separable functions, the penalty will play a decisive role in guiding the direction of optimization, because in noiseless environments, the linkage measurement will be infinitely close to 0 in fully separable functions, so even if in noisy environments, the penalty should still mainly lead the direction of optimization. As the number of nonseparable variables increases, the impact of linkage measurement will increase in the linkage measure function. Theoretically, in the fully nonseparable functions, the effect of the penalty can be ignored, but considering that the strength of interactions between different variables are various, and the performance of a single optimizer to optimize the large-scale problems directly will rapidly decline. Therefore, the existence of the penalty will allow our proposal to ignore some weak interactions. we apply $\lambda = [0.1, 0.01, 0.001, 0.0001, 0.00001]$ as the candidate hyperparameters, and we observe the proportion of linkage measurement and penalty on 15 test functions. Fig. 4 shows the proportion of linkage measurement and penalty in the optimum.

As described in Section IV-A, $f_1^N(x)$ to $f_3^N(x)$ are fully separable functions, $f_4^N(x)$ to $f_{11}^N(x)$ are partially separable functions, and $f_{12}^N(x)$ to $f_{15}^N(x)$ are functions with overlapping subcomponents, and our hypothesis can be mainly satisfied when $\lambda = 0.01$. Therefore, in this paper, we apply $\lambda = 0.01$ as the parameter of our experiments.

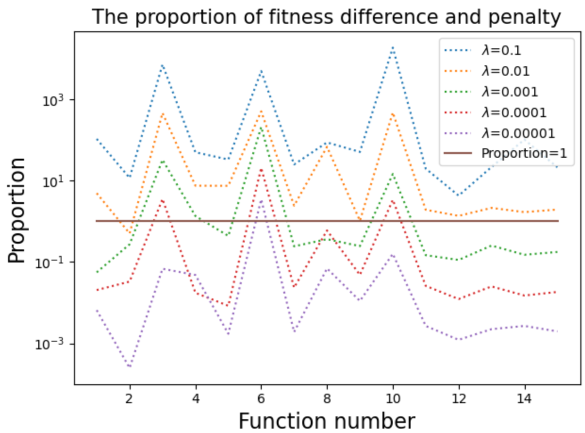


Fig. 4. The proportion of linkage measurement and penalty in 15 functions

C. Performance of EASDECC-LINC-Rmin

In this section, the performance of EASDECC-LINC-Rmin is studied, both on the decomposition of our proposal and the introduction of EAS. Experiments are conducted on the benchmark functions presented in Section IV-A1, and the convergence curve of 25 independent runs is introduced in Fig. 5

We also provide the mean and standard deviation of the optimum at the end of optimization with 25 trial runs in Table IV. The best solution is in bold.

Meanwhile, Table V shows the FEs consumed in decomposition stage for Delta grouping (D), Differential grouping (DG), Variable interaction learning (VIL) and EGALINC-Rmin in noisy environments.

D. Analysis

In this Section, we will analyze the effect of the decomposition strategy in our proposal and the EAS on optimization.

1) EGALINC-Rmin in noisy environments: As we describe in Section II-C, the traditional decomposition methods based on perturbations will fail to detect the interactions in noisy environments, both convergence curves and FEs consumed in the decomposition stage have consistency with analysis. In addition, the decomposition method of our proposal is better than the comparing methods in most test functions. Next, we will explain the reasons why our proposal is better than traditional decomposition methods through two aspects.

First, our proposed decomposition method perturbs on multiple samples, which makes our proposal more robust than the traditional perturbation-based decomposition methods on large-scale problems, although it will increase the consumption of FEs on the decomposition stage. Second, unlike the traditional perturbation-based decomposition methods, it is unnecessary for our proposal to check the ε while identifying the interactions, which means our proposal is not sensitive to the accuracy of sampling. This is the fundamental reason for our proposal that can be applied in noisy environments.

In our decomposition optimization, we employ EGA as the optimizer, so when the strong interactions are identified, the improvement of the linkage measure function is greater, and the fitness of the individual containing the strong interactions is also better, and this individual has a higher possibility of being retained intact to the next generation as an elite individual, which is the feature of EGA.

Although the existence of noise will affect the strength of interactions between variables, EGALINC-Rmin still has a higher probability to identify the relatively stronger interactions. As a simple example, we use $S(x_i, x_j)$ to represent the strength of the interaction between the x_i and x_j. Assuming that $S(x_1, x_2) = 99$, $S(x_2, x_3) = 100$, it is obvious that $S(x_1, x_2) < S(x_2, x_3)$ in noiseless environments, but it is possible that $S^N(x_1, x_2) > S^N(x_2, x_3)$ in noisy environments. However, this situation only occurs when the strengths are close. When $S(x_1, x_2) = 1$, $S(x_2, x_3) = 100$, $S^N(x_1, x_2) > S^N(x_2, x_3)$ is impossible under the $\beta \sim N(0, 0.01)$ noise. Therefore, even in noisy environments, our proposal can still identify relatively strong interactions between variables. It can say that our proposal of decomposition has a good anti-noise ability. Meanwhile, while strong interactions are identified, some weak interactions are ignored. Although it will increase the error in the optimization stage, it can exponentially reduce the search space, which accelerates the convergence of optimization, especially under the FEs limitation.

2) The introduction of explicit averaging: The introduction of fitness-based explicit averaging with dynamic sampling accelerates the optimization in $f_1 - f_3, f_6, f_8, f_{10}, f_{12}$ and f_{15} with significance, which shows the ability of anti-noise. However, it is still a rigorous challenge to handle the noise in large-scale optimization problems with limited FEs. Currently, the popular anti-noise methods are often combined with a large number of re-evaluation and mathematical methods to estimate the real sample fitness. When the number of re-evaluation times is too small, estimation fitness still has a great difference from real fitness, and when the number of re-evaluation times is too large, the computational resources allocated to the optimization will be reduced. Therefore, designing a more efficient anti-noise strategy and balancing the computational resources between anti-noise and optimization is a meaningful research topic.

V. Conclusion and Future Works

In this paper, we propose a novel decomposition method that regards the variable grouping problem as a combinatorial optimization problem, and design the linkage measure function based on LINC-R. According to our previous research, this linkage measure function is multimodal, we attach a penalty to lead the direction of optimization. Besides, we normalize the linkage measurement term in linkage measure function and apply λ to balance the intensity of the linkage measurement and the penalty. Under the background that the perturbation-based decomposition methods will fail to detect the interactions, our proposal can detect the strong interactions while some weak

TABLE IV

OPTIMIZATION RESULTS OF EASDECC-D, EASDECC-VIL, EASDECC-DG, EASDECC-LINC-RMIN, AND DECC-LINC-RMIN

Func	EASDECC-VIL		EASDECC-D		EASDECC-DG		EASDECC-G		EASDECC-LINC-Rmin		DECC-LINC-Rmin	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
f_1	1.57e+11	1.52e+10	5.75e+10	1.08e+10	1.56e+11	6.72e+09	4.80e+10	1.09e+10	**2.89e+09**	**4.62e+08**	1.51e+10	1.01e+09
f_2	7.05e+04	2.98e+03	4.61e+04	1.89e+03	7.11e+04	1.55e+03	4.51e+04	1.79e+03	**2.98e+04**	**2.18e+03**	3.91e+04	9.05e+02
f_3	2.11e+01	1.07e-01	2.11e+01	9.03e-02	2.11e+01	8.45e-02	2.11e+01	9.63e-02	**2.11e+01**	**8.61e-02**	2.12e+01	8.25e-02
f_4	2.73e+12	8.91e+11	1.02e+12	3.51e+11	2.27e+12	7.13e+11	7.98e+11	2.29e+11	4.66e+11	1.17e+11	**3.17e+11**	**9.84e+10**
f_5	2.20e+07	2.04e+06	1.35e+07	1.73e+06	2.23e+07	1.49e+06	1.18e+07	1.46e+06	1.18e+07	1.30e+06	**1.10e+07**	**8.59e+05**
f_6	1.051e+06	5.16e+03	1.050e+06	5.38e+03	1.051e+06	5.41e+03	1.049e+06	6.14e+03	**1.049e+06**	**2.51e+03**	1.051e+06	3.65e+03
f_7	1.62e+12	1.01e+12	6.57e+09	2.83e+09	1.66e+12	1.08e+12	6.09e+09	3.00e+09	2.42e+09	5.41e+08	**1.40e+09**	**6.99e+08**
f_8	8.23e+16	4.46e+16	2.04e+16	1.33e+16	1.37e+17	6.45e+16	8.79e+15	4.72e+15	**6.13e+15**	**1.88e+15**	1.02e+16	4.26e+15
f_9	1.63e+09	1.50e+08	1.10e+09	9.20e+07	1.60e+09	1.34e+08	1.03e+09	1.06e+08	8.94e+08	7.63e+07	**8.84e+08**	**6.08e+07**
f_{10}	9.22e+07	5.77e+05	9.20e+07	6.15e+05	9.23e+07	5.17e+05	**9.20e+07**	**5.76e+05**	9.31e+07	6.29e+05	9.42e+07	4.63e+05
f_{11}	2.90e+14	1.86e+14	1.08e+12	3.97e+11	4.78e+14	3.32e+14	1.12e+12	5.68e+11	4.08e+11	1.40e+11	**2.37e+11**	**1.49e+11**
f_{12}	5.21e+12	2.47e+11	2.44e+12	2.78e+11	5.42e+12	1.99e+11	2.29e+12	3.55e+11	**9.97e+08**	**2.92e+08**	3.08e+11	1.89e+10
f_{13}	2.03e+14	1.11e+14	2.75e+11	9.89e+10	2.72e+14	1.33e+14	2.72e+11	1.07e+11	**2.62e+10**	**4.32e+09**	2.77e+10	3.53e+09
f_{14}	3.27e+14	2.31e+14	2.61e+12	9.98e+11	4.75e+14	2.62e+14	2.42e+12	8.15e+11	4.52e+11	1.38e+11	**3.35e+11**	**1.03e+11**
f_{15}	4.76e+15	1.52e+15	3.80e+12	1.32e+13	4.37e+15	7.76e+14	2.43e+11	3.95e+11	**6.68e+07**	**9.58e+06**	1.39e+08	3.02e+07

TABLE V

THE FES CONSUMED IN DECOMPOSITION OF DELTA GROUPING, DIFFERENTIAL GROUPING, VARIABLE INTERACTION LEARNING AND EGALINC-RMIN

Func	D	DG	VIL	EGALINC-Rmin
f_1	30000	3972	4104	470995
f_2	30000	3972	4480	480525
f_3	30000	3974	4188	475520
f_4	30000	3972	4574	477400
f_5	30000	3972	4524	478255
f_6	30000	3972	4580	478380
f_7	30000	3972	4618	483555
f_8	30000	3972	4230	479285
f_9	30000	3972	4330	479285
$f_1 0$	30000	3972	4106	478945
$f_1 1$	30000	3972	4462	476705
$f_1 2$	30000	3972	4944	482410
$f_1 3$	30000	3972	4516	482540
$f_1 4$	30000	3972	4240	480280
$f_1 5$	30000	3972	4114	483140

interactions are ignored, which performed well on benchmark functions.

Besides, In the optimization stage, we notice that the noise will mislead the direction of optimization, and we also introduce a fitness-based explicit averaging with dynamic sampling to reduce the effect of noise. To evaluate our proposal of decomposition, we compare our proposal with EASDECC-VIL, EASDECC-D, EASDECC-G, and EASDECC-DG on CEC2013 LSGO Suite in noisy environments, we also compare the EASDECC-LINC-Rmin with DECC-LINC-Rmin to verify the efficiency of the introduction of EAS. Experimental results show that in the decomposition stage, our proposal is significantly better than the second-best algorithm in most of the test functions. The introduction of the EAS to the optimizer can also reduce the influence of noise and correct the direction of optimization.

However, we performed extensive pre-experiments to manually define the λ when balancing the intensity of the penalty. In future research, it is one of our research directions to adaptively search the propoer λ by learning the features of the fitness landscape such as reinforcement learning and machine learning. Furthermore, according to the definition of the standard error of mean fitness in EAS, a large number of re-evaluation times are necessary to reduce the standard error, which is contradictory in large-scale optimization problems under the limitation of FEs. In future research, we will introduce the surrogate model for approximating the noise such as the neural network to save the FEs consumed in linkage measure value correction. Finally, our proposal is a promising study for solving large-scale optimization problems in noisy environments.

VI. ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number JP20K11967.

REFERENCES

[1] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, and J. Xu, “An improved differential evolution algorithm and its application in optimization problem,” *Soft Computing*, vol. 25, no. 7, pp. 5277–5298, 2021.

[2] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” in *2005 IEEE congress on evolutionary computation*, vol. 2. IEEE, 2005, pp. 1785–1791.

[3] D. Greiner, J. J. Aznarez, O. Maeso, and G. Winter, “Single- and multi-objective shape design of y-noise barriers using evolutionary computation and boundary elements,” *Advances in Engineering Software*, vol. 41, no. 2, pp. 368–378, 2010.

[4] E. J. Hughes, “Evolutionary multi-objective ranking with uncertainty and noise,” in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2001, pp. 329–343.

[5] J. Li, Q. Zhou, H. Williams, H. Xu, and C. Du, “Cyber-physical data fusion in surrogate- assisted strength pareto evolutionary algorithm for phev energy management optimization,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4107–4117, 2022.

[6] D. Sudholt, “On the robustness of evolutionary algorithms to noise: refined results and an example where noise helps,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 1523–1530.

[7] J.-S. Kim, U.-C. Jeong, D.-W. Kim, S.-Y. Han, and J.-E. Oh, “Optimization of sirocco fan blade to reduce noise of air purifier using a metamodel and evolutionary algorithm,” *Applied Acoustics*, vol. 89, pp. 254–266, 2015.

[8] L. Painton and U. Diwekar, “Stochastic annealing for synthesis under uncertainty,” *European Journal of Operational Research*, vol. 83, no. 3, pp. 489–502, 1995.

[9] J. Diaz and J. Handl, “Implicit and explicit averaging strategies for simulation-based optimization of a real-world production planning problem,” *Informatica (Slovenia)*, vol. 39, pp. 161–168, 01 2015.

[10] W. A. Albukhanajer, J. A. Briffa, and Y. Jin, "Evolutionary multi-objective image feature extraction in the presence of noise," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1757–1768, 2014.

[11] Y. Akimoto, S. Astete-Morales, and O. Teytaud, "Analysis of runtime of optimization algorithms for noisy functions over discrete codomains," *Theoretical Computer Science*, vol. 605, pp. 42–50, 2015.

[12] Y.-W. Chen, Q. Song, X. Liu, P. S. Sastry, and X. Hu, "On robustness of neural architecture search under label noise," *Frontiers in Big Data*, vol. 3, 2020.

[13] C. Qian, J.-C. Shi, Y. Yu, K. Tang, and Z.-H. Zhou, "Subset selection under noise," *Advances in neural information processing systems*, vol. 30, 2017.

[14] M. Köppen, "The curse of dimensionality," in *5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1, 2000, pp. 4–8.

[15] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, Tech. Rep., 1994.

[16] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational optimization and applications*, vol. 21, no. 1, pp. 5–20, 2002.

[17] P. Moscato *et al.*, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, p. 1989, 1989.

[18] E. Li, H. Wang, and F. Ye, "Two-level multi-surrogate assisted optimization method for high dimensional nonlinear problems," *Applied Soft Computing*, vol. 46, pp. 26–36, 2016.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] L. Holmstrom and P. Koistinen, "Using additive noise in back-propagation training," *IEEE transactions on neural networks*, vol. 3, no. 1, pp. 24–38, 1992.

[27] J. M. Sancho, M. San Miguel, S. Katz, and J. Gunton, "Analytical and numerical studies of multiplicative noise," *Physical Review A*, vol. 26, no. 3, p. 1589, 1982.

[28] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of north american fuzzy information processing*. Ieee, 1996, pp. 519–523.

[29] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[30] M. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," 2010, pp. 1–8.

[31] M. Munetomo and D. E. Goldberg, "Linkage identification by non-monotonicity detection for overlapping functions," *Evolutionary computation*, vol. 7, no. 4, pp. 377–398, 1999.

[32] 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.

[33] Y. Wu, X. Peng, H. Wang, Y. Jin, and D. Xu, "Cooperative Coevolutionary CMA-ES with Landscape-Aware Grouping in Noisy Environments," *IEEE Transactions on Evolutionary Computation*, 2022.

[34] G. Gopalakrishnan, B. S. Minsker, and D. E. Goldberg, "Optimal sampling in a noisy genetic algorithm for risk-based remediation design," *Journal of Hydroinformatics*, vol. 5, pp. 11–25, 2001.

[35] P. Rakshit, A. Konar, and S. Das, "Noisy evolutionary optimization algorithms – A comprehensive survey," *Swarm and Evolutionary Computation*, vol. 33, pp. 18–45, 2017.

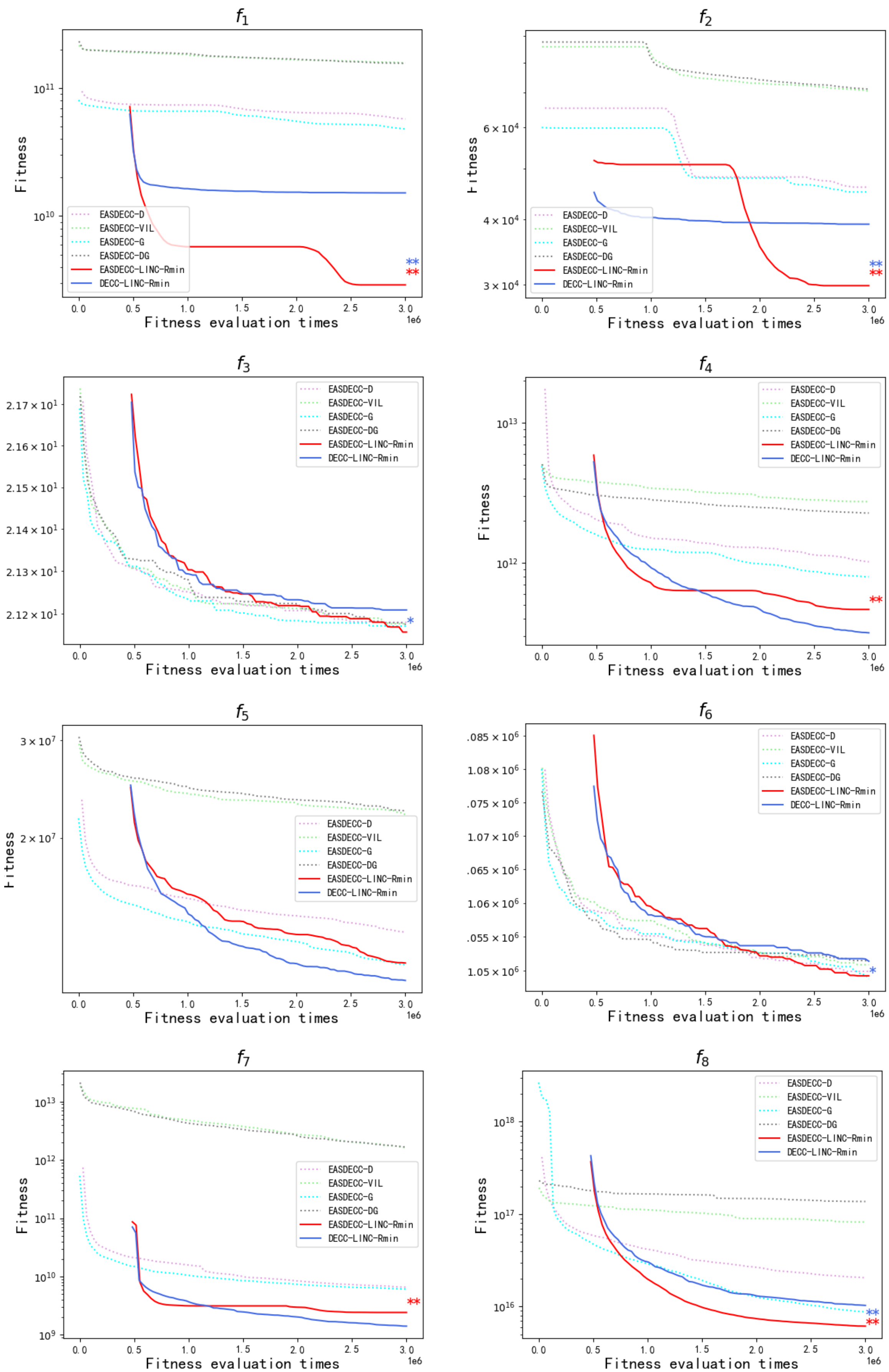
[36] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[37] M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, "DG2: A Faster and More Accurate Differential Grouping for Large-Scale Black-Box Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 929–942, 2017.

[38] R. Zhong and M. Munetomo, "Random Population-based Decomposition Method by Linkage Identification with Non-linearity Minimization on Graph," in *Transactions on Computational Science & Computational Intelligence*. Springer, Accepted.

[39] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems.* University of Michigan, 1975.

[40] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2010, pp. 300–309.



f_1
f_2
f_3
f_4
f_5
f_6
f_7
f_8
Fitness
Fitness evaluation times
1e6
EASDECC-D
EASDECC-VIL
EASDECC-G
EASDECC-DG
EASDECC-LINC-Rmin
DECC-LINC-Rmin

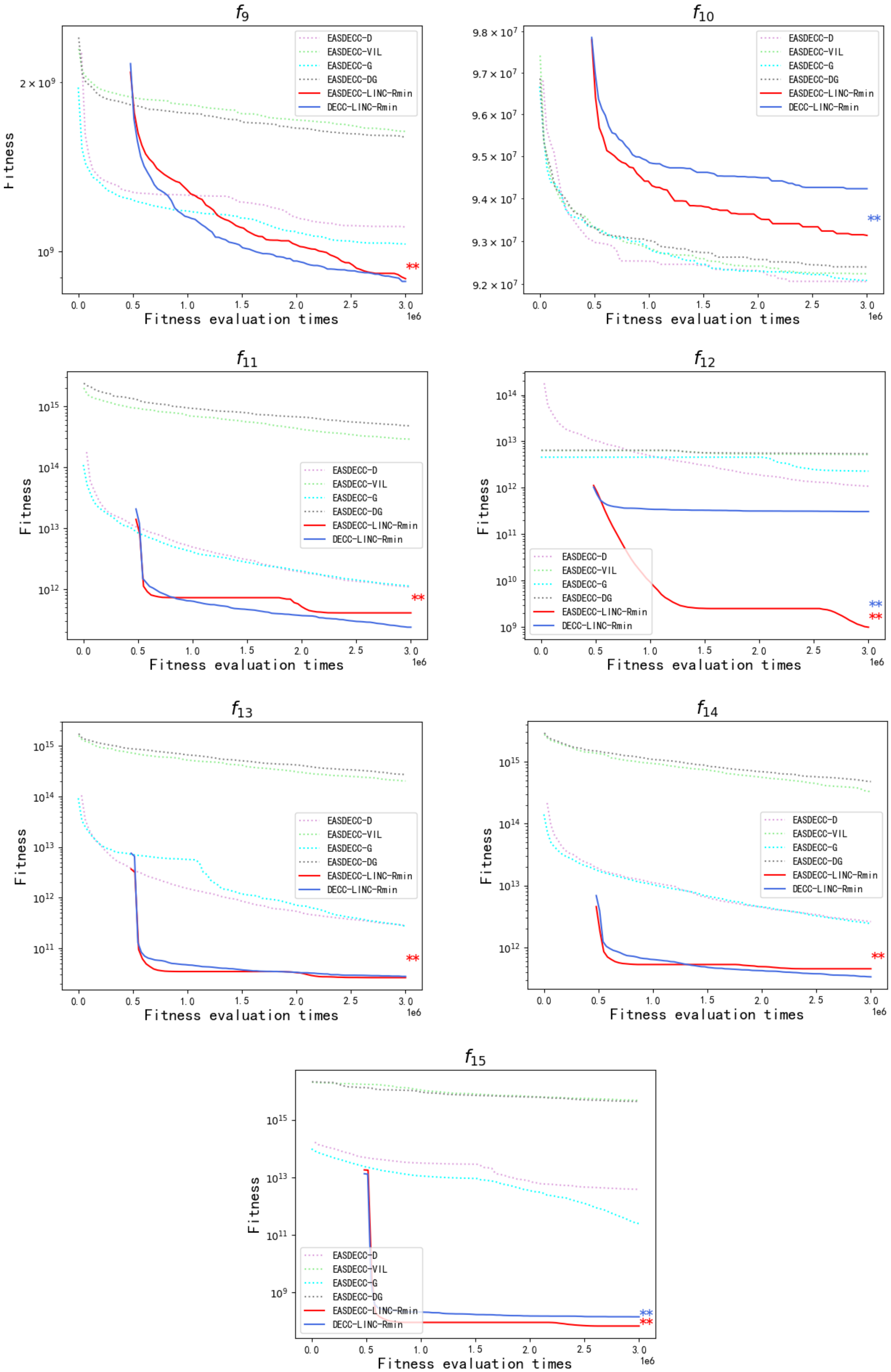


Fig. 5. The convergence curve of EASDECC-D, EASDECC-VIL, EASDECC-DG, EASDECC-LINC-Rmin, and DECC-LINC-Rmin. The gap in the initial period is FEs comsumed for decomposition.