

Received April 15, 2021, accepted May 16, 2021, date of publication May 20, 2021, date of current version July 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3082202

# An Adaptive Mechanism With Cooperative Coevolution and Covariance for Differential Evolution

**BIN WANG**<sup>1</sup>, **LU REN**<sup>1</sup>, **JUAN DIEGO PRADO**<sup>1</sup>, **WENJUAN HE**<sup>2</sup>,  
**HAIYAN JIN**<sup>1</sup>, **QIAOYONG JIANG**<sup>1</sup>, AND **XIAOFAN WANG**<sup>1</sup>

<sup>1</sup>Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

<sup>2</sup>Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China

Corresponding author: Haiyan Jin (jinhaiyan@xaut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Project 61803301 and Project 61976177, in part by the Technology Innovation Guidance Project of Shaanxi Province under Grant 2020CGXNG-026, and in part by the Doctoral Foundation of the Xi'an University of Technology under Grant 112-451116017.

**ABSTRACT** Differential evolution (DE) is an evolutionary algorithm widely used to solve optimization problems with different characteristics in fields where actions and decisions depend on numerical data such as engineering, economics, and logistics. In this paper, an adaptive differential evolution mechanism with cooperative co-evolution and covariance (A-CC/COV-DE) is proposed to overcome the low efficiency of differential evolution when solving large-scale numerical optimization problems, especially when the correlation between the variables of the problem is unknown. An unknown correlation of variables hinders DE from achieving an optimal search process since different types of correlations ideally require distinct optimization strategies. According to the separability of variables, the appropriate evolutionary strategy is selected adaptively. For separable functions, cooperative coevolution is adopted. After using extended differential grouping to split the problem, the sub-components are optimized by differential evolution. This reduces the dimensionality and complexity of the problem, improving its convergence speed and global search ability. For non-separable functions, a covariance matrix is calculated, and then the eigenvector is used to rotate the coordinate system. This leads to eliminate the correlation between variables and improve the search efficiency of differential evolution. We evaluated the performance of A-CC/COV-DE on the CEC 2014 test suite and compared it with state-of-the-art differential evolution algorithms. The experimental results show that our proposal is quite competitive with recent algorithms.

**INDEX TERMS** Cooperative coevolution, covariance, differential evolution, adaptive mechanism.

## I. INTRODUCTION

Differential evolution (DE) is a random search algorithm for numerical optimization problems inspired by natural species evolution [1]. In DE, a mutant vector is generated according to a mutation strategy, and then an experimental vector is generated according to the crossover operator [2]. Finally, the mutated vector is compared with the experimental vector, and the individuals with better fitness function value are reserved for the next generation. Since DE has an important influence in the field of evolutionary computing, researchers have proposed many improvement strategies for DE. For example, Sun *et al.* [3] adopted a novel Gaussian mutation

operator and a modified common mutation operator to collaboratively produce new mutant vectors. In [4], they also proposed an elite representative based individual adaptive regeneration framework (EIR) that can be incorporated into any DE variant easily. In [5], Sun *et al.* proposed a time-varying strategy-based DE algorithm (TVDE), a novel simple variant of DE. In [6], Deng *et al.* designed an adaptive dimension level adjustment framework (ADLA) to relieve the premature convergence or stagnation problem faced by the DE algorithm. In [7], an elite regeneration framework for differential evolution (ERG-DE) was proposed, where a new individual is produced from the search space around each elite individual by sampling Gaussian or Cauchy probability models. In [8], Zhang and Sanderson proposed an adaptive differential evolution with an optional external archive (JADE).

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko<sup>1</sup>.

However, variable interactions in objective functions (i.e. whether the variable is separable or not) has an important influence on evolutionary operators [9]. The operator of the original difference algorithm has rotational variability, and the evolution efficiency will be affected when optimizing the function of variable correlation. In [10], the CoBiDE (differential evolution based on covariance matrix learning and bimodal distribution parameter setting) algorithm proposed by Wang *et al.*, combines DE with a crossover operator based on eigenvector to form a new DE algorithm with rotation invariance. When some correlations between variables exist, the covariance matrix of the sample points can be used to describe them. The feature vector and eigenvalue are used to map the sample points into the new space to eliminate the correlation between variables.

In [11], Potter *et al.* proposed a framework named cooperative coevolution (CC) combined with the classic genetic algorithm as an evolutionary approach to function optimization. In [12], cooperative coevolutionary differential evolution (CCDE) was proposed to combine CC with Differential Evolution for solving large-scale optimization problems, the optimization problem is decomposed into several independent subcomponents, and then these subcomponents are optimized at the same time. The experiments show that it has a good effect when solving problems with uncorrelated variables.

According to the CC framework, if variables are grouped randomly, the optimization process for some variables may interfere with the correct evolution of other variables. Ideally, only interacting decision variables should be assigned to the same subcomponent.

In reference [13], Yao *et al.* proposed the differential grouping method (DG), which was added to the CC framework. This new strategy detects and assigns the relevant decision variables to the same subcomponent. Since DG can only detect direct correlations between variables but cannot detect indirect correlations, the decomposition degree of the decomposition method is relatively low in some test functions. On this basis, an extended differential grouping (XDG) method is proposed in [14], which can decompose the variables of optimization problems appropriately. By embedding XDG into the CC framework, the optimization problems can be decomposed into several subcomponents, where variables are related if they belong to the same subcomponent and are not related if otherwise. Therefore, this paper proposes an adaptive mechanism with cooperative coevolution and covariance for differential evolution. The main contributions are as follows:

(1) After analysis, we found that cooperative coevolution is effective when solving problems that separable variables, but its performance in problems with only non-separable variables is relatively poor. Differential evolution with covariance can analyze the characteristics of samples and rotate the original coordinates according to the rotation invariance feature vector to eliminate

the correlation between variables and improve the algorithm's performance.

- (2) In the case of different correlation of variables, the CC framework and the rotation method based on covariance and eigenvector are analyzed, respectively.
- (3) A new learning mechanism based on coevolution and covariance is proposed, which automatically chooses between the differential evolution algorithm based on the CC framework (CCDE) and the differential evolution algorithm based on covariance (COVDE). Under the premise of unknown correlation of objective function variables, comparable experimental results are obtained.

The structure of the paper is as follows. Section II presents the background for this work. Section III presents an adaptive mechanism with cooperative coevolution and covariance for differential evolution, called A-CC/COV-DE, which is used in these experiments. In Section IV experimental results of the A-CC/COV-DE algorithm on benchmark functions are presented. Section V concludes this paper with some final remarks.

## II. RELATED WORK

### A. DIFFERENTIAL EVOLUTION

Differential evolution is a population-based metaheuristic algorithm [15]. Similar to other numerical optimization algorithms, DE uses a set of real parameter vectors  $x_i = (x_1, \dots, x_D)$ ,  $i = 1, 2, \dots, N$  where D is the dimension of the optimization problem and N is the population size. At the beginning of the search, the individuals in the population are randomly initialized, and the mutant vector  $v_{i,G}$  is generated in the evolution process, the test vector  $u_{i,G}$  is generated by the crossover operation. Finally, individuals with better fitness values are retained in the  $G$  generation according to the selection. The three key steps in Differential Evolution are mutation, crossover and selection:

- (1) Mutation: this operation uses the target vector  $x_{i,G}$  to generate a mutant vector  $v_{i,G}$ .

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \quad (1)$$

The subscripts  $r1$ ,  $r2$  and  $r3$  are randomly selected from  $[1, N]$ , but should be different from  $i$ , the parameter  $F$  is a positive scaling factor.

- (2) Crossover: this operation uses the mutated vector  $v_{i,G}$  and the target vector  $x_{i,G}$  to generate a test vector  $u_{i,G}$ . The common crossover operation of DE is binomial crossover:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } \text{rand}(0, 1) \leq CR \text{ or } j = j_{rand} \\ x_{j,i,G}, & \text{otherwise} \end{cases} \quad (2)$$

where  $\text{rand}(0, 1)$  represents a uniform random number between  $(0, 1)$ ,  $j_{rand}$  is the subscript of decision variables selected uniformly and randomly from  $[1, D]$ , CR is the crossover rate.

(3) Selection: DE uses greedy selection, which chooses the individual with the better fitness between the target vector  $x_{i,G}$  and the test vector  $u_{i,G}$ :

$$x_{G+1} = \begin{cases} u_{i,G}, & \text{if } f(u_{i,G}) < f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \quad (3)$$

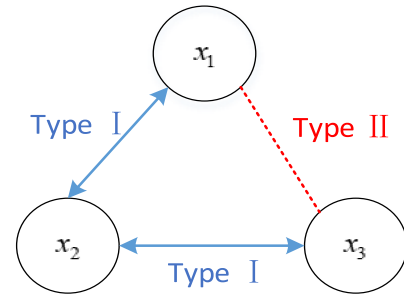
The operation of DE is relatively simple, and it can quickly find optimal solutions in low-dimensional problems. However, it becomes poor in high-dimensional problems. Therefore, to solve this problem, the framework of cooperative coevolution is proposed.

**B. COOPERATIVE COEVOLUTION**

Cooperative coevolution (CC) is often used to solve large-scale optimization problems. Many researchers have done a lot of work on CC [16]–[22], and it has been widely used in various fields [23]–[25]. CC decomposes a high-dimensional optimization problem into multiple low-dimensional subproblems and then cyclically optimizes those multiple subproblems [26]–[28]. As the name implies, the difficulty of cooperative coevolution lies in the method for decomposing the optimization problem. There are several decomposition techniques, such as static grouping [29], random grouping [12], [30] and grouping strategies based on variable interactions [31].

The most common decomposition method is random grouping, that is, the decision variables are randomly assigned to subcomponents in each evolution cycle. From the mathematical point of view, the probability of dividing two interacting decision variables into the same group in several evolution cycles is quite large, but when the number of interactive decision variables is greater than 2, the random grouping becomes uncontrollable and the evolution performance is getting poorer. Delta grouping is similar to random grouping [32], this method decomposes a high-dimensional problem into several low-dimensional subproblems. It is necessary to pre-define the decomposition of decision variables into k-groups. If there are many interacting variables in the optimization problem, a large k-value will affect the performance of the algorithm negatively. However, for a smaller number of interacting variables, a small k-value will reduce the performance of the algorithm. In short, once the value of k is determined, the decision variables will be decomposed into k groups of a fixed size, which may be disadvantageous to practical optimization problems.

Differential grouping (DG) is an automatic decomposition strategy [13], which allocates decision variables to subcomponents, and is derived from the definition of partial additive separable functions. DG algorithm checks the interaction between each of the variables. If there is an interaction between two variables, both are placed into the same subcomponent; if a variable does not interact with any other variable, it is considered a separable variable. DG method can detect simple direct variable interactions but it performs poorly when identifying complex interactions. Therefore,



**FIGURE 1. Two types of variable interaction.**

an extended differential grouping (XDG) method was proposed to solve the shortcomings found in DG.

Extended differential grouping can capture two types of interaction between variables. As shown in Figure 1, the variables of Type I interact directly, eg.  $x_1$  and  $x_2$  (or  $x_2$  and  $x_3$ ) interact directly, and the variables of Type II interact indirectly, eg.  $x_1$  and  $x_3$  are linked by  $x_2$ . The formal definition of interacting types is listed below:

*Definition 1:* In an objective function  $f(\vec{X})$ , variables  $x_i$  and  $x_j$  interact directly with each other if  $\exists$  a candidate solution  $\vec{x}_*$ , s.t.

$$\frac{\partial f}{\partial x_i \partial x_j} \Big|_{\vec{x}_*} \neq 0 \quad (4)$$

denoted by  $x_i \leftrightarrow x_j$ . Variables  $x_i$  and  $x_j$  interact indirectly with each other if for all candidate solutions,

$$\frac{\partial f}{\partial x_i \partial x_j} = 0 \quad (5)$$

and  $\exists$  a set of variables  $\{x_{k1}, \dots, x_{km}\} \subset \vec{X}$ , s.t.  $x_i \leftrightarrow x_{k1} \leftrightarrow \dots \leftrightarrow x_{km} \leftrightarrow x_j$ . Variables  $x_i$  and  $x_j$  are independent with each other if for all candidate solutions, (5) holds and  $\nexists$  a set of variables  $\{x_{k1}, \dots, x_{km}\} \subset \vec{X}$ , s.t.  $x_i \leftrightarrow x_{k1} \leftrightarrow \dots \leftrightarrow x_{km} \leftrightarrow x_j$ .

XDG decomposes the optimization problem into several subproblems according to the correlation between decision variables of the objective function [14]. Algorithm 1 shows the pseudo-code of the XDG method, which is mainly divided into three stages. The first stage is to determine direct interactions between variables in lines 3-25. It executes pairwise comparisons once between each decision variable to detect the interactions. The detection works by evaluating a series of vectors with small perturbations of each pair of variables. The difference in fitness values is checked to decide if the variables are dependent or independent of each other. The second stage is aimed to identify indirect interactions between variables in lines 26-33. It simply searches for overlaps between the resulting groups from the first stage and merges the groups with common variables. If two sub-components have the same decision variable, they will be merged until all the sub-components do not intersect. The third stage is to group all separable variables into the same subcomponent in lines 34-40. It is performed by merging every group that contains

only one decision variable into a single group. The decision variables contained in each subproblem are interactive and non-separable, but each group of subproblems can be separated.

Algorithm 2 shows the pseudo-code of the CC framework. The algorithm is mainly divided into two stages: the grouping stage and the optimization stage. The grouping stage is performed by XDG (Algorithm 1). The optimization stage optimizes the subcomponents formed in the grouping phase. For simplicity, the optimization process uses differential evolution. However, even if XDG is added to the CC framework, optimizing non-separable intragroup variables remains a problem. Therefore, an adaptive selection mechanism with two different methods is proposed. The first method is to add XDG to the CC framework, and then use the DE mutation operation to optimize; the other method is to add covariance to the DE process for optimization, which can make up for the limited performance of CC in the optimization process of non-separable problems.

### C. COVARIANCE

It is usually difficult to determine the optimal search direction in the evolution process by using Differential Evolution by itself, especially when the variables are related. In this section, we use the covariance matrix to analyze the features of the sample points. Additionally, we use the eigenvector to transform the points in the original Cartesian coordinate system to a new coordinate system and eliminate the correlation between variables according to the rotation invariance [33].

According to the best  $N/2$  individuals in the population after differential evolution, the covariance matrix is calculated:

$$\text{cov}(\mathbf{P}_{1:N/2}) = [\text{cov}(i, j)]_{D \times D} \quad (6)$$

where  $\text{cov}(i, j)$  is the covariance of the  $i$ th and the  $j$ th dimensions of the first  $N/2$  individuals in the current population, which is calculated as:

$$\text{cov}(k, j) = \frac{1}{N/2 - 1} \sum_{i=1}^{N/2} (x_{i,k} - \bar{x}_k)(x_{i,j} - \bar{x}_j) \quad (7)$$

$$k = 1, 2, \dots, D; \quad j = 1, 2, \dots, D$$

$\text{cov}(\mathbf{P}_{1:N/2})$  is decomposed as follows:

$$\text{cov}(\mathbf{P}_{1:N/2}) = \mathbf{R} \mathbf{\Lambda}^2 \mathbf{R}' \quad (8)$$

where  $\mathbf{R}$  is the  $D \times D$  orthogonal matrix coordinate system representing the features, and each row of  $\mathbf{R}$  is the covariance matrix  $\text{cov}(\mathbf{P}_{1:N/2})$ ,  $\mathbf{R}'$  represents the transformation from the characteristic coordinate system to the original coordinate system, and  $\mathbf{\Lambda}$  is a diagonal matrix composed of eigenvalues.

An individual  $x_i$  in the original Cartesian coordinate system is expressed in the feature coordinate system as follows:

$$x'_i = x_i \mathbf{R} \quad (9)$$

Using the search equation of differential evolution, the candidate solution  $v'_j$  is generated in the characteristic

### Algorithm 1 XDG Algorithm

---

```

1: Input:  $f$ : objective function
2:    $D$ : dimension
3:    $\vec{ub}$ : upper bounds of decision variables
4:    $\vec{lb}$ : lower bounds of decision variables
5:    $\varepsilon$ : the threshold to identify direct interaction between
      decision variables.
6:  $IM \leftarrow \text{zeros}(D - 1, D)$  //  $IM$ : Interaction Matrix
7:  $sep\_var \leftarrow []$  //  $sep\_var$ : separable variables
8: for  $i = 1$  to  $D$  do
9:    $groups(i) = \{i\}$ 
10:   $\vec{X}_1 \leftarrow \vec{lb}$ 
11:   $\vec{X}_2 \leftarrow \vec{X}_1$ 
12:   $\vec{X}_2(i) \leftarrow \vec{ub}(i)$ 
13:   $\Delta_1 \leftarrow f(\vec{X}_1) - f(\vec{X}_2)$ 
14:  for  $j = i + 1$  to  $D$  do
15:    if  $IM(i, j) = 0$  then
16:       $\vec{X}_1(j) \leftarrow (\vec{ub}(j) + \vec{lb}(j))/2$ 
17:       $\vec{X}_2(j) \leftarrow (\vec{ub}(j) + \vec{lb}(j))/2$ 
18:       $\Delta_2 \leftarrow f(\vec{X}_1) - f(\vec{X}_2)$ 
19:      if  $|\Delta_1 - \Delta_2| > \varepsilon$  then
20:         $IM(i, j) \leftarrow 1$ 
21:         $groups(i) \leftarrow groups(i) \cup \{j\}$ 
22:      end if
23:    else
24:       $groups(i) \leftarrow groups(i) \cup \{j\}$ 
25:    end if
26:  end for
27: for  $p, q \in groups(i)$  &  $p < q$  do
28:    $IM(p, q) \leftarrow 1$ 
29: end for
30: end for
31: while the number of variables in groups is not  $D$  do
32:  for  $p, q \in \{1 : \text{num}(groups)\}$  &  $p < q$  do
33:    if  $groups(p) \cap groups(q) \neq \emptyset$  then
34:       $groups(p) \leftarrow groups(p) \cup groups(q)$ 
35:      delete  $groups(q)$ 
36:    end if
37:  end for
38: end while
39: for  $i = 1$  to  $\text{num}(groups)$  do
40:  if  $\text{length}(groups(i)) = 1$  then
41:     $sep\_var \leftarrow sep\_var \cup group(i)$ 
42:    delete  $groups(i)$ 
43:  end if
44: end for
45:  $groups \leftarrow groups \cup \{sep\_var\}$ 
46: Output:  $groups$ 

```

---

coordinate system:

$$v'_{i,j} = x'_{r1,j} + F \cdot (x'_{r2,j} - x'_{r3,j}) \quad (10)$$

where  $r1, r2$  and  $r3$  are randomly selected from  $[1, N]$ . The candidate solutions in the characteristic coordinate system are then transformed into the original Cartesian coordinate

**Algorithm 2** CC Framework

```

1: Input: N: population size.
2:   D: dimension.
3:   F: scaling factor
4:   CR: crossover rate
5:   FESmax: maximum number of function evaluations
6:  $groups \leftarrow grouping(f, D, lb, ub, \epsilon)$  // grouping stage
7:  $pop \leftarrow rand(popsize, N)$  // optimization stage
8:  $(best, best\_val) \leftarrow min(f(pop))$ 
9: for  $i \leftarrow 1$  to number of generations do
10: for  $j \leftarrow 1$  to size(groups) do
11:    $indices \leftarrow groups[j]$ 
12:    $subpop \leftarrow pop[:, indices]$ 
13:    $subpop \leftarrow optimizer(best, subpop, FES)$ 
14:    $pop[:, indices] \leftarrow subpop$ 
15:    $(best, best\_val) \leftarrow min(func(pop))$ 
16: end for
17: end for
18: Output: take the current best solution as optimal solution
output.
```

system to be evaluated against the current population:

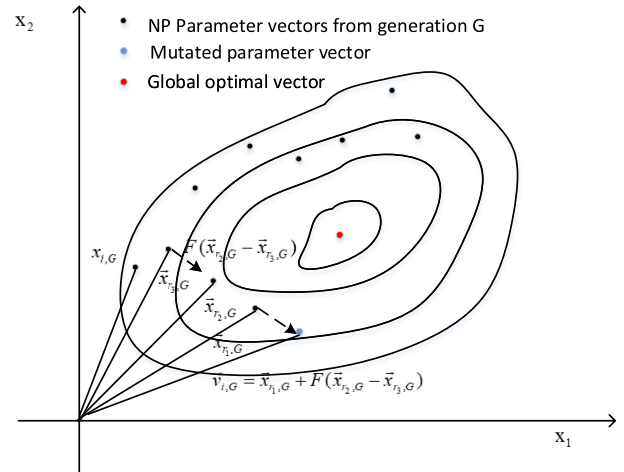
$$v_i = v'_i \mathbf{R}' \tag{11}$$

**III. A-CC/COV-DE**

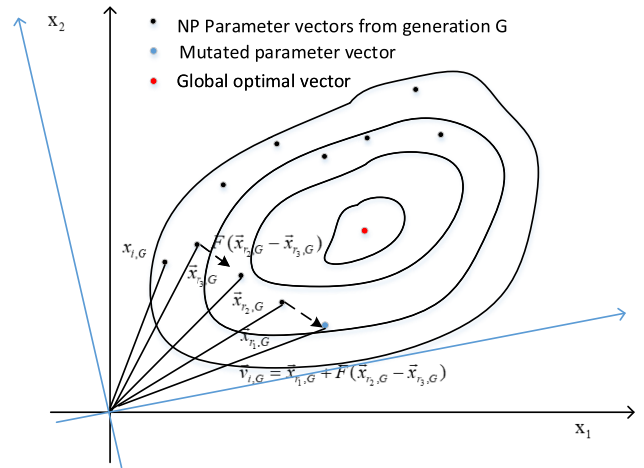
**A. MOTIVATION**

Differential evolution has clear advantages in some optimization problems, but it is known that the performance of differential evolution in large-scale optimization problems will get more complicated. The most popular methods to solve large-scale optimization problems are based on grouping and dimensionality reduction, with the CC framework being one of these approaches. We observe that the performance of the cooperative coevolution algorithm is better for optimization problems with separable variables, but the performance of the CC algorithm is relatively poor in optimization problems with non-separable variables. For example, if a 10-dimensional problem is fully separable, CC will decompose the individuals into 10 one-dimensional ones, and then each portion of the original individuals are solved separately. When the dimensionality increases, CC can accelerate the convergence speed of the algorithm when solving this kind of problems. For non-separable problems, CC will split the individuals, allocating related variables into the same subcomponent. Therefore, during the evolution, the convergence speed and global search ability of the algorithm will be affected, especially when the dimensionality is increasing.

Figure 2 shows the process of generating a mutant vector  $v_{i,G+1}$  in the two-dimensional cost function. As shown in Figure 3, the construction of the characteristic coordinate system is used to release the correlation between variables. In the characteristic coordinate system, the mutant vector is closer to the global optimal solution, which helps to improve the convergence of the algorithm.



**FIGURE 2.** An example of the process of generating  $v_{i,G+1}$  in the two-dimensional cost function.



**FIGURE 3.** The search equation of DE in the Eigen coordinate system.

In differential evolution, the covariance matrix is added to analyze the characteristics of sample points. According to the eigenvector, the points in the original Cartesian coordinate system are transformed into the feature coordinate system to eliminate the correlation between variables. Our experiments show that for the optimization problem with non-separable variables, using covariance to analyze variables in differential evolution can improve the performance of the algorithm. For the problems with separable variables, the performance of the algorithm is poor. In practical optimization problems, it is difficult to know beforehand whether variables can be completely separated or not. Therefore, to improve the optimization performance of the algorithm, we propose an adaptive evolutionary mechanism based on cooperative coevolution and covariance.

**B. ALGORITHM FRAMEWORK**

The flow chart of the proposed algorithm is shown in Figure 4, consists of two parts. The left part is cooperative coevolution

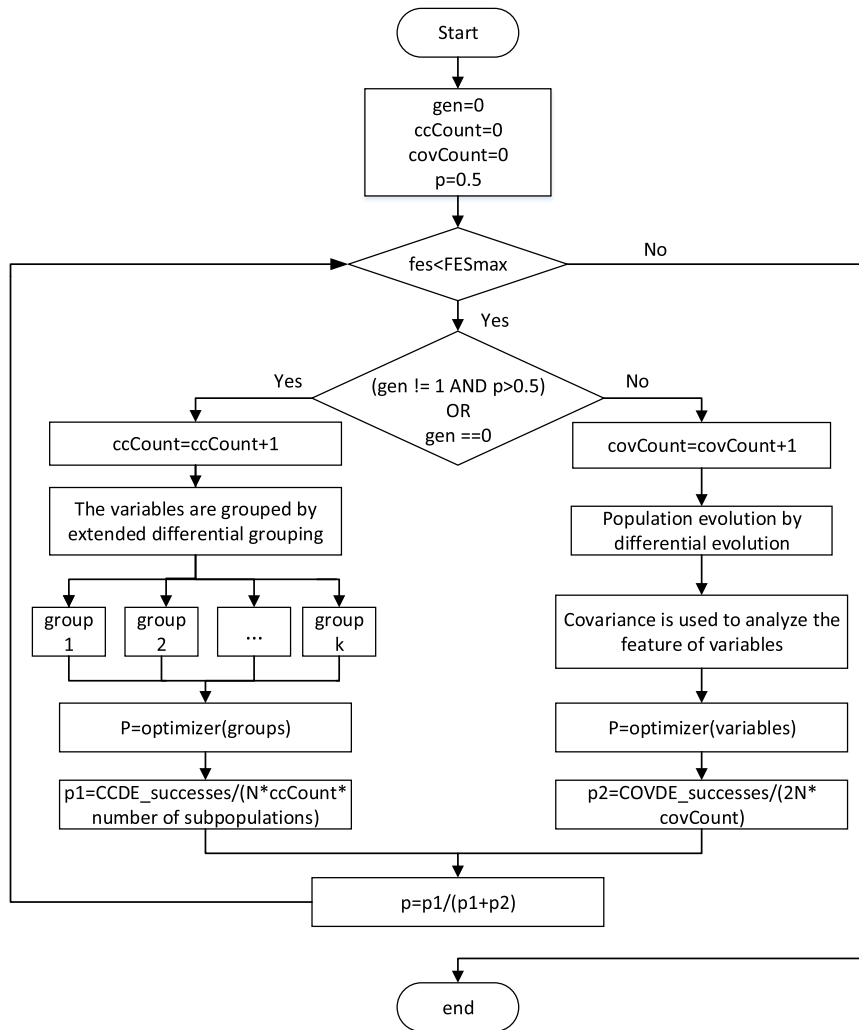


FIGURE 4. Flow chart of the proposed algorithm.

with differential evolution as optimizer (CCDE). Firstly, in the framework of CC, extended differential grouping (XDG) is used to group variables according to the correlation between the variables, and then differential evolution is used to evolve the sub-components after grouping in a cyclic manner. The part on the right is COVDE. Firstly, differential evolution is used to evolve the population, and then covariance is used to analyze the characteristics of variables (to describe the correlation between variables). DE operator with covariance has rotation invariance. According to the characteristic vector of samples, the points in the coordinate system are transformed into a characteristic coordinate system, which eliminates the correlation between variables and is conducive to improving the performance when evolving non-separable problems. Finally, the two algorithms are adaptively selected according to the success rate of each evolutionary algorithm (the ratio between the generated offspring that is successfully selected for the next generation in all generations and the total number of functions evaluations used by each algorithm), so that the proposed method can

have better optimization performance for variable dependent or unrelated optimization problems.

### C. AN ADAPTIVE EVOLUTIONARY SELECTION MECHANISM

According to the experimental observations, using XDG along with CC and differential evolution, has a good effect on problems with separable variables, but the results of functions with partially separable as well as non-separable variables are far from ideal, especially when compared with COVDE. To improve the performance of the algorithm, adaptive selection is conducted between CCDE and COVDE. After each generation, a probability  $p$  is calculated as follows:

$$P = \frac{p1}{p1 + p2} \tag{12}$$

where  $P$  represents probability for CCDE as selected for the next generation, which is calculated by its relative success rate. The overall success rate of CCDE and COVDE are depicted as  $p1$  and  $p2$ , respectively, and are calculated

**TABLE 1.** The statistical results (Mean (Std)) of A-CC/COV-DE vs. CCDE, COVDE, DE, ADLADE, CSDE, TVDE over 50 independent runs on the CEC2014 benchmarks with 30D.

Functions	CCDE	COVDE	DE	ADLADE	CSDE	TVDE	A-CC/COV-DE
F1	7.77E+02 + (3.50E+03)	1.49E+05 + (8.04E+04)	5.34E+04 + (4.18E+04)	6.65E+04 + (7.06E+04)	2.30E+05 + (1.58E+05)	1.16E+05 + (1.92E+05)	<b>1.89E-02</b> (1.26E-01)
F2	1.31E-05 + (1.50E-05)	2.10E-14 + (1.38E-14)	1.76E-14 ~ (1.51E-14)	1.65E-14 ~ (1.42E-14)	2.56E-14 + (8.61E-15)	<b>0.00E+00</b> - (0.00E+00)	1.48E-14 (1.43E-14)
F3	6.96E-05 + (5.98E-05)	5.12E-14 + (3.49E-14)	3.00E-10 + (2.12E-09)	7.96E-14 + (8.67E-14)	4.89E-14 + (1.99E-14)	<b>2.05E-14</b> ~ (2.76E-14)	<b>1.36E-14</b> (2.45E-14)
F4	9.52E+00 ~ (2.06E+01)	1.05E+01 ~ (2.38E+01)	9.22E+00 - (2.16E+01)	3.18E+00 - (1.24E+01)	4.87E+00 ~ (1.58E+01)	<b>1.26E+00</b> - (5.47E+00)	9.87E+00 (2.25E+01)
F5	2.10E+01 + (5.36E-02)	2.07E+01 - (6.74E-02)	2.10E+01 + (6.45E-02)	2.04E+01 - (1.89E-01)	<b>2.02E+01</b> - (8.29E-02)	2.05E+01 - (6.90E-02)	2.09E+01 (6.41E-02)
F6	<b>1.06E-01</b> ~ (2.26E-01)	5.14E+00 + (7.16E+00)	<b>8.14E-01</b> ~ (1.05E+00)	7.64E-01 + (6.92E-01)	1.47E+00 + (1.01E+00)	2.64E-01 + (6.27E-01)	<b>1.73E-01</b> (3.86E-01)
F7	3.59E-01 + (1.07E-01)	1.67E-03 - (4.49E-03)	6.90E-04 - (2.44E-03)	1.63E-03 - (3.84E-03)	1.86E-08 - (8.88E-08)	<b>0.00E+00</b> - (0.00E+00)	1.53E-01 (1.31E-01)
F8	5.14E-13 + (1.20E-13)	5.27E+01 + (8.88E+00)	1.26E+01 + (4.03E+00)	1.50E+01 + (4.28E+00)	7.06E+00 + (2.14E+00)	3.81E+00 + (1.69E+00)	<b>4.21E-13</b> (2.02E-13)
F9	3.96E+01 + (5.13E+00)	6.41E+01 + (9.98E+00)	1.02E+02 + (6.75E+01)	<b>2.59E+01</b> - (8.25E+00)	3.22E+01 ~ (8.68E+00)	2.72E+01 - (9.02E+00)	3.15E+01 (7.97E+00)
F10	<b>3.04E-02</b> ~ (2.35E-02)	2.91E+03 + (4.01E+02)	3.71E+02 + (6.84E+02)	1.19E+02 + (1.28E+02)	1.15E+02 + (1.12E+02)	7.93E+01 + (8.66E+01)	<b>2.71E-02</b> (2.46E-02)
F11	1.92E+03 ~ (1.79E+02)	3.57E+03 + (3.04E+02)	6.60E+03 + (3.75E+02)	<b>1.34E+03</b> - (5.74E+02)	2.72E+03 + (4.94E+02)	2.80E+03 + (6.53E+02)	1.91E+03 (3.71E+02)
F12	2.61E+00 + (2.88E-01)	1.06E+00 - (1.66E-01)	2.17E+00 + (4.84E-01)	6.52E-01 - (2.93E-01)	<b>3.41E-01</b> - (9.97E-02)	6.83E-01 - (1.42E-01)	1.28E+00 (1.98E-01)
F13	3.43E-01 + (4.55E-02)	3.18E-01 ~ (4.95E-02)	2.95E-01 ~ (3.52E-02)	2.51E-01 - (4.42E-02)	<b>1.57E-01</b> - (3.69E-02)	2.27E-01 - (3.48E-02)	3.01E-01 (5.67E-02)
F14	3.45E-01 + (6.54E-02)	2.34E-01 - (3.04E-02)	2.87E-01 - (7.29E-02)	2.67E-01 - (5.51E-02)	<b>1.75E-01</b> - (4.09E-02)	2.40E-01 - (5.00E-02)	3.06E-01 (3.82E-02)
F15	1.55E+01 + (1.20E+00)	7.16E+00 ~ (1.05E+00)	1.42E+01 + (1.82E+00)	6.17E+00 - (2.71E+00)	<b>3.76E+00</b> - (1.37E+00)	6.01E+00 - (1.78E+00)	7.48E+00 (1.06E+00)
F16	1.24E+01 + (2.81E-01)	1.17E+01 ~ (3.11E-01)	1.22E+01 + (2.50E-01)	1.02E+01 - (7.38E-01)	1.07E+01 - (6.27E-01)	<b>9.98E+00</b> - (6.46E-01)	1.17E+01 (3.23E-01)
F17	8.49E+03 + (1.40E+04)	4.02E+02 + (1.75E+02)	2.77E+03 + (3.89E+03)	3.43E+03 + (3.59E+03)	2.09E+03 + (1.22E+03)	8.72E+03 + (1.97E+04)	<b>2.74E+02</b> (1.13E+02)
F18	4.07E+01 + (5.90E+00)	2.35E+01 + (8.77E+00)	1.29E+01 + (1.25E+01)	5.15E+01 + (8.77E+00)	2.17E+01 + (8.57E+00)	7.53E+02 + (1.14E+03)	<b>1.17E+01</b> (2.64E+00)
F19	<b>2.99E+00</b> ~ (2.65E-01)	4.70E+00 + (9.79E-01)	<b>3.61E+00</b> ~ (1.31E+00)	<b>3.13E+00</b> ~ (8.17E-01)	4.32E+00 + (6.44E-01)	4.12E+00 + (7.32E-01)	<b>3.08E+00</b> (6.62E-01)
F20	2.33E+01 + (4.68E+00)	1.91E+01 + (7.13E+00)	1.16E+01 - (4.34E+00)	2.13E+01 + (1.40E+01)	<b>1.04E+01</b> - (3.53E+00)	1.19E+01 ~ (4.18E+00)	1.28E+01 (2.81E+00)
F21	4.20E+02 + (1.31E+02)	1.68E+02 + (1.38E+02)	<b>1.06E+02</b> ~ (9.43E+01)	2.17E+02 + (2.59E+02)	2.99E+02 + (1.43E+02)	5.15E+02 + (3.01E+02)	<b>1.06E+02</b> (6.25E+01)
F22	2.97E+02 + (1.52E+02)	1.16E+02 ~ (8.31E+01)	5.75E+01 - (7.37E+01)	5.25E+01 + (5.44E+01)	<b>2.77E+01</b> - (1.99E+01)	1.22E+02 ~ (8.26E+01)	1.11E+02 (8.55E+01)
F23	<b>3.15E+02</b> ~ (3.98E-13)	<b>3.15E+02</b> ~ (4.47E-13)	<b>3.15E+02</b> ~ (3.98E-13)	<b>3.15E+02</b> ~ (4.02E-13)	<b>3.15E+02</b> ~ (4.02E-13)	3.15E+02 + (4.14E-13)	<b>3.15E+02</b> (4.02E-13)
F24	2.24E+02 ~ (3.99E+00)	2.25E+02 ~ (2.15E+00)	2.26E+02 ~ (5.02E+00)	2.26E+02 ~ (3.97E+00)	<b>2.04E+02</b> - (8.69E+00)	2.23E+02 + (4.04E+00)	2.26E+02 (4.41E+00)
F25	2.03E+02 ~ (3.76E-01)	2.03E+02 ~ (4.49E-01)	2.03E+02 ~ (4.80E-01)	2.03E+02 ~ (6.44E-01)	2.03E+02 + (2.17E-01)	2.04E+02 + (4.87E-01)	<b>2.03E+02</b> (4.55E-01)
F26	1.00E+02 + (4.77E-02)	1.00E+02 ~ (5.36E-02)	1.00E+02 ~ (4.60E-02)	1.00E+02 - (4.00E-02)	<b>1.00E+02</b> - (3.28E-02)	<b>1.00E+02</b> - (4.16E-02)	1.02E+02 (1.41E+01)
F27	3.48E+02 ~ (4.55E+01)	3.84E+02 ~ (3.39E+01)	3.46E+02 ~ (4.57E+01)	3.46E+02 ~ (4.49E+01)	3.38E+02 ~ (4.64E+01)	<b>3.22E+02</b> - (3.81E+01)	3.67E+02 (4.67E+01)
F28	8.20E+02 - (4.43E+01)	8.66E+02 ~ (4.27E+01)	8.19E+02 ~ (5.84E+01)	8.21E+02 - (3.60E+01)	7.90E+02 - (2.26E+01)	<b>7.87E+02</b> - (3.79E+01)	8.53E+02 (4.36E+01)
F29	8.69E+02 + (1.63E+02)	<b>6.01E+02</b> ~ (2.22E+02)	7.86E+02 + (1.41E+02)	7.91E+02 + (1.52E+02)	9.69E+02 + (1.24E+02)	1.33E+03 + (2.77E+02)	<b>6.64E+02</b> (1.67E+02)
F30	1.05E+03 + (6.47E+02)	<b>7.40E+02</b> ~ (3.54E+02)	1.10E+03 + (5.79E+02)	9.96E+02 + (4.92E+02)	1.05E+03 + (3.45E+02)	1.91E+03 + (7.38E+02)	<b>7.47E+02</b> (4.38E+02)
	19/2/9	13/4/13	14/5/11	12/12/6	13/13/4	14/13/3	

as follows:

$$p1 = \frac{\omega1}{N \times n1 \times k} \tag{13}$$

$$p2 = \frac{\omega2}{2 \times N \times n2} \tag{14}$$

where  $\omega1$  and  $\omega2$  represent the number of times CCDE and COVDE were successful in the evolutionary process, respectively.  $n1$  and  $n2$  represent the evolution times of CCDE and COVDE, respectively, and  $k$  is the number of XDG groups. The selection probability calculation does not depend

**TABLE 2.** The p values for Wilcoxon rank-sum tests of A-CC/COV-DE vs. CCDE, COVDE, DE, ADLADE, CSDE, TVDE over 50 independent runs on the CEC2014 benchmarks with 30D.

Functions	CCDE	COVDE	DE	ADLADE	CSDE	TVDE
F1	1.94E-15	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F2	2.09E-18	3.27E-02	3.70E-01	5.51E-01	3.16E-05	3.79E-09
F3	7.43E-19	6.24E-09	9.89E-12	1.31E-11	5.79E-10	1.94E-01
F4	1.11E-01	8.82E-01	2.11E-02	1.33E-03	6.03E-01	7.56E-07
F5	1.29E-13	1.20E-11	3.02E-10	1.95E-17	7.07E-18	7.07E-18
F6	1.27E-01	2.26E-10	7.26E-01	1.77E-02	8.05E-14	1.23E-07
F7	5.64E-11	1.05E-13	1.00E-14	1.35E-13	4.24E-13	1.25E-19
F8	1.98E-02	6.25E-18	6.14E-18	6.23E-18	6.22E-18	6.25E-18
F9	1.00E-06	1.08E-17	2.07E-05	6.36E-04	7.64E-01	4.97E-03
F10	3.47E-01	6.61E-18	6.61E-18	6.61E-18	6.61E-18	6.61E-18
F11	3.91E-01	8.46E-18	7.07E-18	1.47E-07	1.27E-12	1.85E-10
F12	7.07E-18	7.56E-07	2.65E-13	6.52E-15	7.07E-18	1.63E-17
F13	1.44E-04	7.14E-02	4.26E-01	5.11E-06	4.46E-17	2.89E-10
F14	1.65E-04	2.05E-13	3.90E-04	6.87E-08	1.20E-16	2.95E-11
F15	7.07E-18	1.67E-01	1.35E-16	9.02E-04	5.33E-16	1.28E-06
F16	2.26E-14	5.01E-01	7.29E-10	1.80E-16	5.24E-15	1.21E-17
F17	8.46E-18	1.28E-04	5.57E-14	4.01E-16	8.99E-17	7.07E-18
F18	7.07E-18	1.99E-12	1.63E-02	2.95E-17	2.81E-11	1.92E-14
F19	5.88E-01	3.46E-14	1.32E-01	6.17E-01	4.21E-15	1.20E-11
F20	6.17E-15	3.69E-07	1.89E-02	1.45E-05	2.48E-04	7.36E-02
F21	2.47E-17	4.23E-02	4.67E-01	1.15E-07	9.46E-12	7.92E-16
F22	4.73E-09	7.28E-01	1.12E-04	3.27E-05	2.32E-09	3.33E-01
F23	3.27E-01	3.27E-01	3.27E-01	NaN	NaN	2.30E-02
F24	3.36E-01	8.44E-01	8.50E-01	3.13E-01	7.50E-18	9.76E-06
F25	6.67E-01	7.12E-01	2.37E-01	2.84E-01	4.30E-02	4.83E-05
F26	3.90E-04	2.99E-01	1.36E-01	1.17E-08	1.29E-17	1.55E-12
F27	1.55E-01	6.19E-02	5.44E-02	5.04E-02	3.06E-01	5.72E-06
F28	2.84E-04	2.29E-01	8.80E-04	2.55E-04	8.19E-12	1.29E-10
F29	2.27E-13	3.54E-01	4.20E-13	3.96E-12	7.07E-18	7.07E-18
F30	4.92E-04	4.67E-01	5.27E-05	2.23E-04	9.02E-07	1.16E-13

on the setting of any parameters, it only depends on the fitness value of the two evolution methods involved in the evolution process, which makes it simple and straightforward. If  $p > 0.5$  at the start of a generation, the success rate of CCDE is greater than COVDE's. Hence, CCDE will be the optimizer for that generation. Otherwise, the evolution will continue with COVDE. The advantage of this method is that whether the optimization problem can be separated or not, it can obtain better optimization results. The pseudocode of the adaptive evolutionary selection mechanism is given by Algorithm 3.

#### D. ALGORITHM COMPLEXITY ANALYSIS

To objectively compare the performance of the algorithms, here we mainly analyze the complexity of the proposed algorithm and differential evolution algorithm. The computational complexity of the algorithm can be divided into two parts: time complexity and space complexity. The spatial complexity of the algorithm is the same as DE, with the addition of  $O(D^2)$  to store the covariance matrix in COVDE, and  $O(N)$  to store the group assigned for each dimension in CCDE.

For the DE algorithm, its time complexity is mainly related to the number of individuals in the population and the problem dimensionality. The population size is  $N$ , and the problem dimensionality is  $D$ . Then, the algorithm has a time complexity of  $O(N \cdot D)$  for a single iteration, being  $O(N \cdot D \cdot \text{Gen})$  for the whole run, where Gen is the number of generations given by FES/ $N$ .

For the proposed algorithm A-CC/COV-DE, CCDE has a complexity of  $O(N \cdot D \cdot S)$ , where  $S$  represents the number of groups calculated by XDG. Since the maximum number of possible sub-populations in a problem is  $D$ , this complexity can also be represented as  $O(N \cdot D^2)$ . For a whole run of the algorithm, the complexity is given by  $O(N \cdot D \cdot S \cdot \text{Gen}/S)$ , therefore it has the same time complexity as DE. The second part of A-CC/COV-DE, which is COVDE, also has a complexity of  $O(N \cdot D^2)$ , whose upper bound is given by the calculation of the covariance matrix.

Since both sections of the algorithm have the same complexity, we can conclude that the overall time complexity of A-CC/COV-DE is  $O(N \cdot D^2)$ .



**TABLE 3.** The statistical results (Mean (Std)) of A-CC/COV-DE vs. CCDE, COVDE, DE, ADLADE, CSDE, TVDE over 50 independent runs on the CEC2014 benchmarks with 50D.

Functions	CCDE	COVDE	DE	ADLADE	CSDE	TVDE	A-CC/COV-DE
F1	8.01E+00 + (5.55E+00)	7.24E+05 + (3.54E+05)	1.43E+06 + (5.16E+05)	1.40E+06 + (5.60E+05)	1.73E+06 + (6.30E+05)	6.70E+05 + (5.12E+05)	<b>2.50E-01</b> (1.39E-01)
F2	4.22E+06 + (1.57E+06)	1.63E+02 - (7.31E+02)	3.02E+00 - (1.19E+01)	3.63E+00 - (9.07E+00)	1.53E+02 - (6.24E+02)	<b>6.75E-09</b> - (2.48E-08)	3.41E+04 (1.95E+04)
F3	1.71E+01 + (9.05E+00)	<b>4.44E-12</b> - (9.10E-12)	2.75E-01 + (9.37E-01)	4.69E-01 + (9.91E-01)	3.06E-01 + (4.14E-01)	6.19E-06 - (2.07E-05)	1.35E-05 (1.53E-05)
F4	9.34E+01 + (1.09E+01)	8.24E+01 ~ (2.21E+01)	<b>5.78E+01</b> - (3.67E+01)	6.27E+01 ~ (3.72E+01)	9.66E+01 + (2.92E+00)	8.96E+01 ~ (1.45E+01)	8.13E+01 (2.10E+01)
F5	2.12E+01 + (3.22E-02)	2.09E+01 - (4.89E-02)	2.11E+01 + (3.64E-02)	2.08E+01 - (1.20E-01)	<b>2.05E+01</b> - (8.48E-02)	2.07E+01 - (8.28E-02)	2.11E+01 (4.20E-02)
F6	3.99E+00 ~ (2.74E+00)	4.44E+01 + (8.62E+00)	1.42E+00 - (1.33E+00)	1.75E+00 - (1.52E+00)	4.94E+00 ~ (2.82E+00)	<b>6.14E-01</b> - (9.49E-01)	4.13E+00 (2.48E+00)
F7	6.79E-01 + (8.17E-02)	1.36E-13 - (4.59E-14)	<b>1.27E-13</b> - (3.73E-14)	1.48E-04 - (1.05E-03)	1.91E-13 - (6.26E-14)	1.82E-13 - (6.08E-14)	3.80E-01 (1.69E-01)
F8	<b>9.48E-13</b> ~ (1.98E-13)	1.37E+02 + (1.27E+01)	1.87E+02 + (4.92E+01)	3.63E+01 + (6.45E+00)	2.03E+01 + (5.45E+00)	1.40E+01 + (3.89E+00)	<b>9.35E-13</b> (2.10E-13)
F9	8.98E+01 + (8.59E+00)	1.63E+02 + (1.77E+01)	3.53E+02 + (1.33E+01)	5.72E+01 - (1.60E+01)	5.59E+01 - (1.13E+01)	<b>4.96E+01</b> - (1.11E+01)	8.79E+01 (2.05E+01)
F10	<b>7.99E-03</b> ~ (8.65E-03)	6.64E+03 + (3.94E+02)	9.19E+03 + (1.47E+03)	8.56E+02 + (4.41E+02)	9.82E+02 + (3.76E+02)	4.53E+02 + (2.72E+02)	<b>6.50E-03</b> (9.52E-03)
F11	<b>4.03E+03</b> ~ (3.32E+02)	7.50E+03 + (4.47E+02)	1.30E+04 + (2.83E+02)	4.63E+03 + (1.40E+03)	5.34E+03 + (9.45E+02)	7.26E+03 + (1.30E+03)	<b>4.07E+03</b> (3.01E+02)
F12	3.42E+00 + (3.06E-01)	1.58E+00 - (2.03E-01)	3.27E+00 + (2.70E-01)	1.50E+00 - (4.14E-01)	<b>6.22E-01</b> - (1.66E-01)	1.08E+00 - (2.67E-01)	1.76E+00 (1.96E-01)
F13	5.50E-01 ~ (9.21E-02)	4.02E-01 - (4.36E-02)	4.61E-01 - (4.49E-02)	3.87E-01 - (6.13E-02)	<b>2.36E-01</b> - (3.13E-02)	3.09E-01 - (4.02E-02)	5.69E-01 (8.95E-02)
F14	3.62E-01 + (1.17E-01)	<b>2.61E-01</b> ~ (2.27E-02)	3.38E-01 + (1.13E-01)	3.64E-01 + (1.75E-01)	2.85E-01 + (1.76E-01)	3.39E-01 + (1.63E-01)	<b>2.79E-01</b> (7.20E-02)
F15	3.25E+01 + (1.59E+00)	1.88E+01 ~ (1.89E+00)	3.10E+01 + (1.27E+00)	2.24E+01 + (4.93E+00)	<b>7.20E+00</b> - (2.15E+00)	1.29E+01 - (3.85E+00)	1.87E+01 (1.80E+00)
F16	2.23E+01 + (2.48E-01)	2.12E+01 ~ (3.93E-01)	2.21E+01 + (2.25E-01)	2.09E+01 - (6.46E-01)	2.02E+01 - (7.11E-01)	<b>1.98E+01</b> - (7.33E-01)	2.12E+01 (3.73E-01)
F17	7.72E+04 + (6.05E+04)	1.29E+03 + (1.97E+02)	1.58E+04 + (1.12E+04)	1.52E+04 + (1.33E+04)	5.04E+04 + (3.29E+04)	3.44E+04 + (3.06E+04)	<b>1.14E+03</b> (2.23E+02)
F18	1.21E+02 + (2.91E+01)	8.91E+01 + (1.81E+01)	1.35E+02 + (7.75E+00)	1.46E+02 + (1.50E+01)	1.30E+02 + (4.32E+01)	4.03E+02 + (4.80E+02)	<b>3.30E+01</b> (4.09E+00)
F19	6.89E+00 + (8.99E-01)	1.14E+01 + (1.23E+00)	1.22E+01 + (8.26E-01)	1.06E+01 + (1.15E+00)	1.13E+01 + (1.03E+00)	1.74E+01 + (9.39E+00)	<b>6.74E+00</b> (1.05E+00)
F20	7.19E+01 + (7.29E+00)	6.04E+01 + (1.01E+01)	9.81E+01 + (1.34E+01)	1.06E+02 + (1.02E+01)	6.74E+01 + (1.79E+01)	1.02E+02 + (4.64E+01)	<b>3.58E+01</b> (5.46E+00)
F21	1.35E+04 + (1.18E+04)	1.05E+03 + (2.07E+02)	2.64E+03 + (6.04E+02)	2.71E+03 + (8.54E+02)	1.20E+04 + (1.03E+04)	4.24E+04 + (7.78E+04)	<b>7.65E+02</b> (1.51E+02)
F22	1.43E+03 + (1.33E+02)	5.54E+02 - (1.45E+02)	7.66E+02 + (3.85E+02)	<b>1.83E+02</b> - (1.73E+02)	3.27E+02 - (1.63E+02)	3.78E+02 - (1.87E+02)	6.25E+02 (1.39E+02)
F23	<b>3.44E+02</b> ~ (4.59E-13)	<b>3.44E+02</b> ~ (4.45E-13)	<b>3.44E+02</b> ~ (4.21E-13)	<b>3.44E+02</b> ~ (4.26E-13)	<b>3.44E+02</b> ~ (4.50E-13)	3.44E+02 + (4.50E-13)	<b>3.44E+02</b> (4.45E-13)
F24	2.70E+02 ~ (2.84E+00)	2.70E+02 ~ (1.88E+00)	2.70E+02 ~ (2.71E+00)	2.70E+02 ~ (3.11E+00)	<b>2.62E+02</b> - (4.39E+00)	2.64E+02 - (3.98E+00)	2.70E+02 (2.22E+00)
F25	2.06E+02 + (5.99E-01)	2.05E+02 ~ (4.86E-01)	2.05E+02 - (4.54E-01)	<b>2.05E+02</b> - (3.71E-01)	2.07E+02 + (7.93E-01)	2.08E+02 + (1.47E+00)	2.06E+02 (4.37E-01)
F26	1.01E+02 - (5.16E-02)	1.02E+02 ~ (1.41E+01)	<b>1.00E+02</b> - (6.19E-02)	1.00E+02 ~ (6.57E-02)	1.00E+02 - (4.60E-02)	1.16E+02 + (4.69E+01)	1.06E+02 (2.39E+01)
F27	3.78E+02 - (4.02E+01)	7.89E+02 ~ (3.96E+02)	3.70E+02 - (3.80E+01)	3.71E+02 - (3.72E+01)	<b>3.47E+02</b> - (2.24E+01)	3.51E+02 - (2.66E+01)	6.89E+02 (3.79E+02)
F28	1.07E+03 - (4.76E+01)	1.35E+03 ~ (7.20E+01)	1.08E+03 - (4.58E+01)	1.08E+03 - (4.39E+01)	1.07E+03 - (2.76E+01)	<b>1.06E+03</b> - (3.53E+01)	1.36E+03 (6.93E+01)
F29	1.20E+03 + (3.40E+02)	<b>7.05E+05</b> ~ (4.98E+06)	9.40E+02 + (2.39E+02)	9.90E+02 + (2.49E+02)	1.60E+03 + (1.48E+02)	1.47E+03 + (3.89E+02)	<b>6.57E+02</b> (1.30E+02)
F30	8.37E+03 ~ (4.03E+02)	8.46E+03 ~ (4.42E+02)	<b>8.25E+03</b> - (3.02E+02)	8.36E+03 ~ (4.33E+02)	8.30E+03 ~ (2.70E+02)	8.59E+03 ~ (5.08E+02)	8.47E+03 (4.66E+02)
+/-/~	19/3/8	11/7/12	18/10/2	13/12/5	14/12/4	14/14/2	

**IV. EXPERIMENTS AND ANALYSIS**

In this section, experimental tests are conducted and their results are presented and discussed.

**A. EXPERIMENTAL SETTINGS**

To evaluate the performance of the proposed algorithm, A-CC/COV-DE is compared with CCDE, COVDE, DE,

**TABLE 4.** The p values for Wilcoxon rank-sum tests of A-CC/COV-DE vs. CCDE, COVDE, DE, ADLADE, CSDE, TVDE over 50 independent runs on the CEC2014 benchmarks with 50D.

Functions	CCDE	COVDE	DE	ADLADE	CSDE	TVDE
F1	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F2	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F3	7.07E-18	6.88E-18	7.07E-18	7.07E-18	7.07E-18	1.23E-09
F4	1.93E-10	4.06E-01	2.74E-03	7.47E-02	7.61E-10	1.75E-01
F5	2.27E-16	7.68E-15	4.65E-13	4.69E-15	7.07E-18	7.07E-18
F6	6.92E-01	4.46E-17	1.00E-07	7.07E-06	3.26E-01	5.15E-13
F7	2.65E-13	5.20E-19	2.12E-19	1.23E-19	1.98E-18	2.11E-18
F8	5.88E-01	6.20E-18	6.20E-18	6.20E-18	6.20E-18	6.20E-18
F9	1.54E-03	8.00E-17	7.07E-18	4.42E-13	2.02E-16	2.07E-17
F10	5.21E-01	6.62E-18	6.62E-18	6.62E-18	6.62E-18	6.62E-18
F11	6.17E-01	7.07E-18	7.07E-18	1.15E-02	3.57E-15	2.95E-17
F12	7.07E-18	1.95E-05	7.07E-18	2.84E-04	7.07E-18	2.85E-16
F13	2.96E-01	1.06E-14	6.68E-10	4.44E-15	7.07E-18	2.95E-17
F14	2.94E-13	5.57E-02	1.69E-10	2.56E-07	5.64E-06	5.19E-08
F15	7.07E-18	9.04E-01	7.07E-18	6.86E-05	1.63E-17	9.01E-12
F16	8.46E-18	7.38E-01	1.84E-17	9.44E-03	6.75E-12	6.32E-16
F17	7.07E-18	5.46E-04	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	3.39E-11
F19	3.02E-02	1.84E-17	7.07E-18	1.27E-16	2.47E-17	8.46E-18
F20	7.07E-18	1.27E-16	7.07E-18	7.07E-18	2.02E-16	6.34E-17
F21	7.07E-18	6.20E-11	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F22	7.07E-18	1.25E-02	4.83E-02	1.05E-15	1.55E-12	3.46E-10
F23	8.22E-02	1.00E+00	1.13E-01	1.87E-01	6.55E-01	7.55E-21
F24	6.94E-01	6.74E-01	8.90E-01	9.12E-01	5.09E-17	3.31E-16
F25	1.04E-05	7.04E-02	2.00E-02	2.08E-03	3.28E-14	3.02E-15
F26	2.95E-12	9.92E-01	6.56E-07	8.82E-01	5.64E-17	4.94E-06
F27	4.29E-05	2.40E-01	2.90E-06	5.11E-06	3.78E-10	3.39E-09
F28	7.07E-18	3.91E-01	7.07E-18	7.07E-18	7.07E-18	7.07E-18
F29	1.63E-17	9.75E-01	1.41E-10	5.42E-13	7.07E-18	7.07E-18
F30	5.15E-01	7.33E-01	2.08E-03	5.67E-02	4.63E-01	1.39E-01

**TABLE 5.** Mean ranks from Friedman tests for each algorithm and dimension.

Algorithm	D=30		D=50		avg
CSDE	3.30	(1)	3.53	(1)	3.42
TVDE	3.35	(2)	3.57	(2)	3.46
A-CC/COV-DE	3.58	(3)	3.65	(3)	3.62
ADLADE	3.60	(4)	3.77	(4)	3.68
DE	4.03	(5)	4.67	(6)	4.35
COVDE	4.97	(6)	3.87	(5)	4.42
CCDE	5.17	(7)	4.95	(7)	5.06

ADLADE, CSDE, TVDE in the 30 test functions provided by the CEC2014 test suite. To further illustrate the effectiveness of A-CC/COV-DE, two different dimensions are used on the tests (30 and 50). The code for A-CC/COV-DE was written in Matlab software.

The properties of the CEC2014 benchmark functions are as follows [11]:

- F1-F3: Unimodal function, non-separable functions
- F4-F16: Simple multimodal function, F8, F10 are separable functions, the remaining are non-separable functions.
- F17-F22: Hybrid functions, non-separable functions
- F23-F30: Composition functions, non-separable functions

The subcomponent optimizer used in this paper is DE, the adaptive crossover rate is 0.9, the scale factor is 0.5, the population size N is (2D), The maximum number of iterations FESmax is set to  $D^*10000$ , and  $D=30, 50$  is the function dimension. In this paper, the experiments are performed independently 50 times.

**B. COMPARISON OF A-CC/COV-DE RESULTS WITH OTHER ALGORITHMS**

To evaluate the performance of the proposed algorithm, this paper uses CCDE, COVDE, DE, ADLADE, CSDE, TVDE in the CEC2014 benchmark suite as a comparison, Kruskal

**Algorithm 3** Adaptive Evolution Selection Mechanism

---

```

1: Input:  $N$ : population size.
2:    $D$ : dimension.
3:    $F$ : scaling factor
4:    $CR$ : crossover rate
5:    $FES_{max}$ : maximum number of function evaluations
6: % Initialization:
7:  $gen = 0$ ;
8:  $ccCount = 0$ ;
9:  $covCount = 0$ ;
10: Create  $N$  solutions  $x_i (i = 1, 2, \dots, N)$  in the search space
    randomly.
11: The population is evaluated according to the objective
    function value.
12: Set the current function evaluations as  $FES = N$ .
13: while  $FES \leq FES_{max}$  do
14:   if ( $p > 0.5 \&\& gen \neq 1$ ) ||  $gen = 0$ 
15:      $ccCount = ccCount + 1$ ;
16:     Group variables according to Algorithm 1.
17:      $pop \leftarrow rand(popsize, N)$ 
18:      $(best, best\_val) \leftarrow min(func(pop))$ 
19:     for  $i \leftarrow 1$  to  $cycles$  do
20:       for  $j \leftarrow 1$  to  $size(groups)$  do
21:          $indicies \leftarrow groups[j]$ ;
22:          $subpop \leftarrow pop[:, indicies]$ ;
23:          $subpop \leftarrow optimizer(best, subpop, usedFES)$ ;
24:          $pop[:, indicies] \leftarrow subpop$ ;
25:          $(best, best\_val) \leftarrow min(func(pop))$ ;
26:         Calculate the overall success rate of CCDE
           by (13);
27:       end for
28:     end for
29:   else
30:      $covCount = covCount + 1$ ;
31:     % Differential mutation operator
32:     Create a mutant vector  $v_{i,G}$  by (1);
33:     % Binomial crossover operator
34:     Create a trial vector  $u_{i,G}$  by (2);
35:     Evaluate the objective function values  $f(u_{i,G})$ ;
36:     % Greedy selection operator
37:     if  $f(u_{i,G}) < f(x_{i,G})$ 
38:        $x_{i,G} = u_{i,G}$ ;
39:     end if
40:   Evaluate the objective function values of individuals
     and sort them.
41:   Calculate the covariance matrix by using the first
      $N/2$  individuals.
42:   Generate the candidate solution  $v'_{i,j}$  in Eigen coordinate
     system by (9 - 10).
43:   Convert  $v'_{i,j}$  to the original coordinate system by (11).
44:   if  $f(v_i) < f(x_i)$ 
45:      $x_i = v_i$ ;
46:   end if

```

---

Wallis rank-sum test was used to obtain the p-value, and Wilcoxon rank-sum test was used to identify the difference

**Algorithm 3** (Continued.) Adaptive Evolution Selection Mechanism

---

```

47:  $usedFES = usedFES + 2 * N$ ;
48: Calculate the overall success rate of COVDE by (14).
49: end
50:  $p = p1/(p1 + p2)$ ;
51: for  $i = 1 : usedFES$ 
52:    $FES = FES + 1$ ;
53: end
54:  $gen = gen + 1$ ;
55: end
56: Output: take the current best solution as optimal solution
    output.

```

---

between the proposed algorithm and the other compared algorithms. The test results of the compared algorithms are marked as “+/-/~”, which means that the marked value is worse, better, or similar to those obtained by A-CC/COV-DE, respectively. The test results for  $D=30$  and  $D=50$  are shown in Table 1 and Table 3, and the average convergence plots for some representative benchmark functions are presented in Figure 5 and Figure 6, respectively. The values highlighted in bold represent the best average results for each benchmark function.

It can be seen from Tables 1 and 3 that the results confirm our initial hypothesis, cooperative coevolution is effective when solving problems that contain separable variables, but its performance in problems with only non-separable variables is relatively poor. For example, in the separable problems F8 and F10, the performance of CCDE is better than COVDE, while in non-separable problems, the performance of the COVDE algorithm is better than CCDE. The cause for this is the ability of differential evolution with covariance to analyze the characteristics of samples and rotate the original coordinates according to the rotation invariance feature vector to eliminate the correlation between variables and improve the algorithm's performance. The results of A-CC/COV-DE algorithm are better than those of CCDE, COVDE and DE in most functions, which means that the adaptive selection of the appropriate evolutionary method based on the correlation of variables can improve the efficiency of the algorithm. For  $D=30$ , the experimental results of our algorithm are similar to ADLADE, CSDE and TVDE. For  $D=50$ , A-CC/COV-DE performs better than ADLADE, CSDE and TVDE. This means that as dimensionality increases, A-CC/COV-DE show better signs of robustness and scalability when solving separable problems as well as non-separable problems, especially in functions F17, F18, F19, F20. Hence, these results show some competitive advantages compared with other algorithms.

Tables 2 and 4 show the p values for the Wilcoxon rank-sum tests of A-CC/COV-DE vs. CCDE, COVDE, DE, ADLADE, CSDE, TVDE over 50 independent runs on the CEC2014 benchmarks with 30D and 50D.

All the algorithms in the experiments were also compared by the Friedman test carried out on the medians of

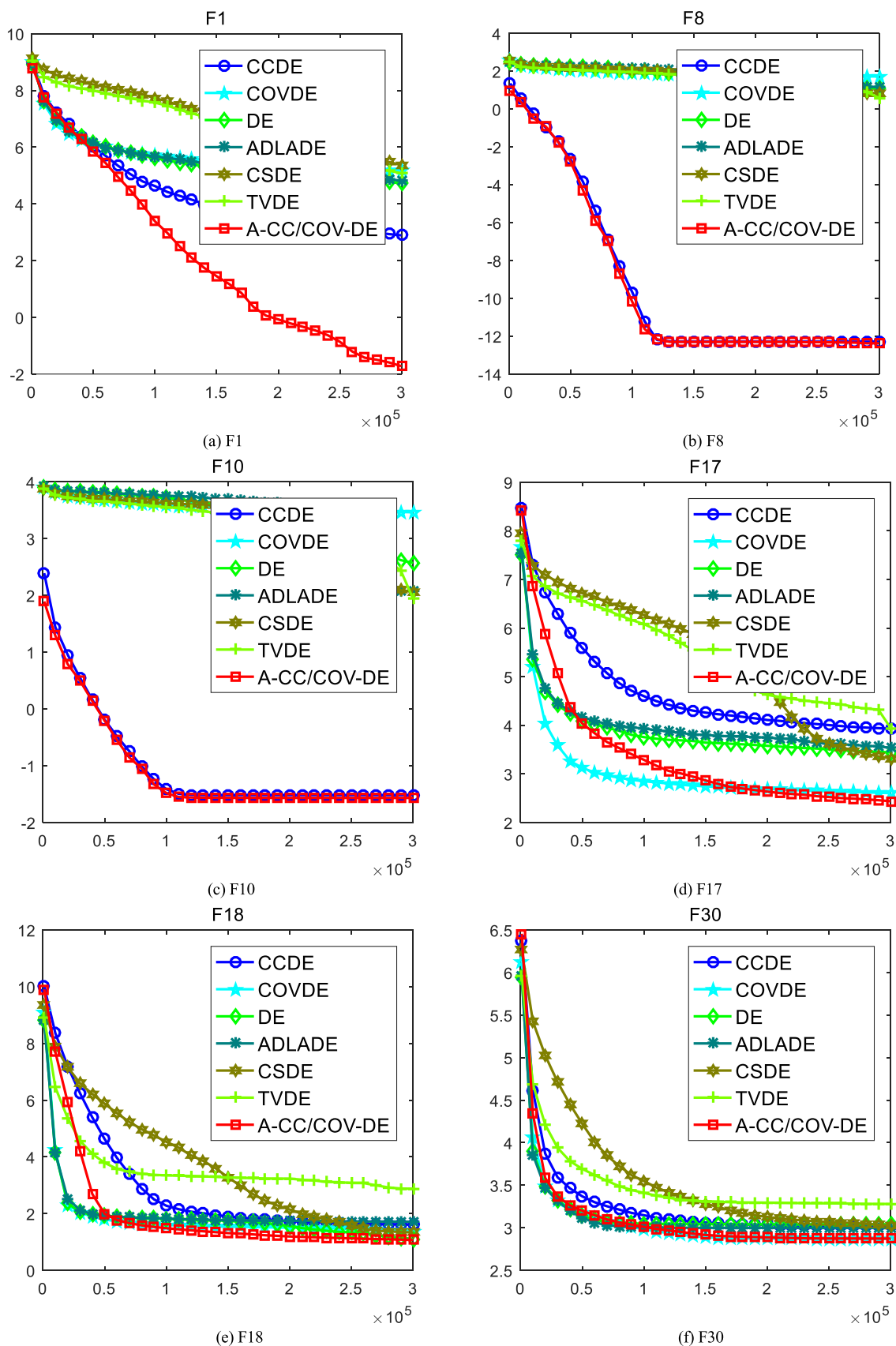
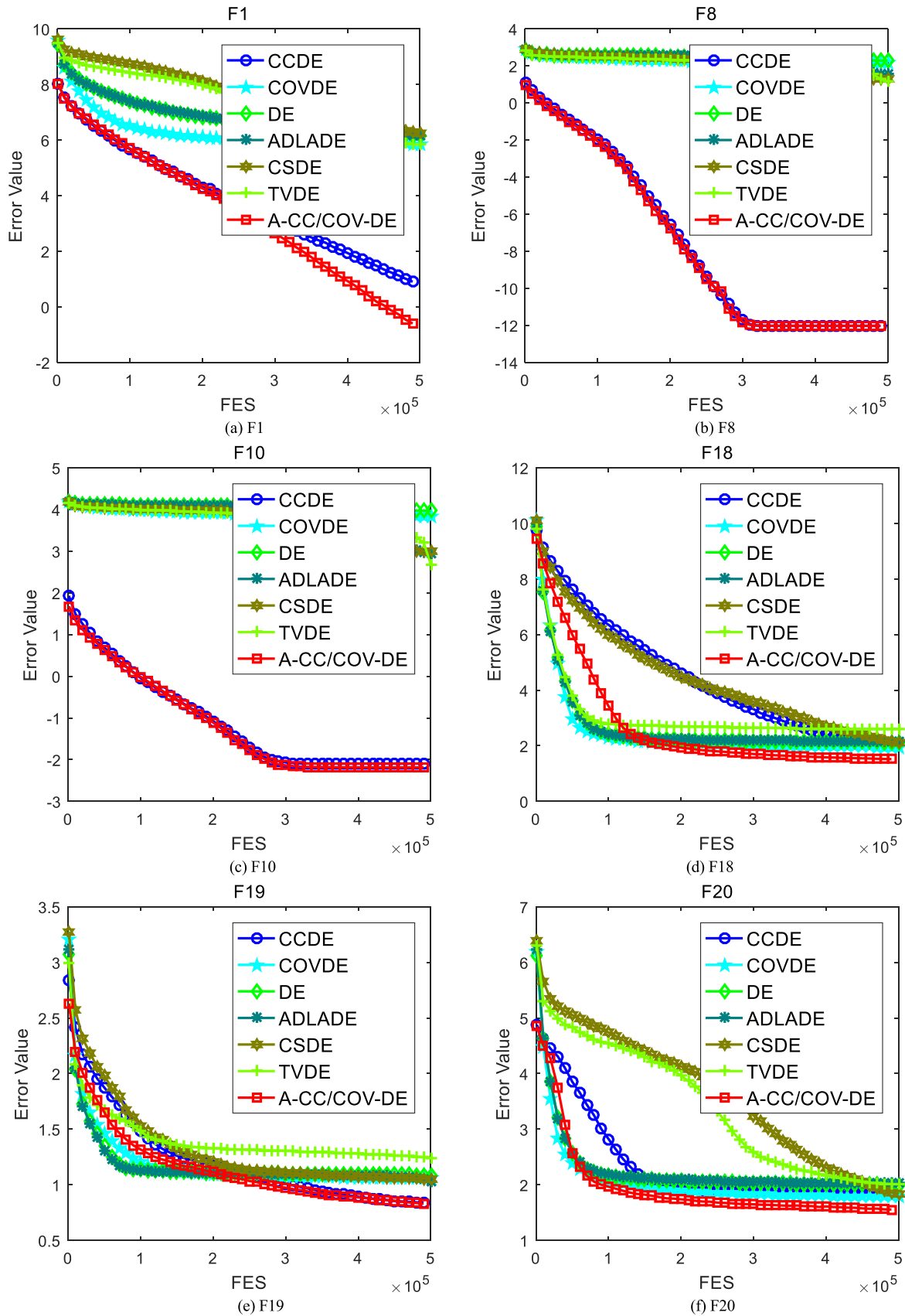


FIGURE 5. The mean function error values versus numbers of function evaluations for the compared algorithms over 50 independent runs (30D).



**FIGURE 6.** The mean function error values versus numbers of function evaluations for the compared algorithms over 50 independent runs (50D).

the minimal function value errors in each dimension. The average ranks of the algorithms according to the Friedman test are shown in Table 5. Considering the average mean ranks, we can see that CSDE and TVDE show a better overall performance, indicating a possible good balance to solve a wide range of problems. However, as can be seen from Table 1 and Table 3, A-CC/COV-DE has the largest number of wins among the 30 benchmark problems, depicted in bold typeface in the tables. This acts as an indicator of the strong ability of our algorithm for converging towards the global optimum. The performance is especially remarkable in hybrid functions with complex landscapes and a set of different mixed properties.

## V. CONCLUSION

Over the past many years, researchers have made a lot of exploration on optimization problems. As a numerical optimization method, differential evolution is popular in optimization problems due to its simplicity, but it is difficult to calculate the optimal solution for high-dimensional and complex optimization problems. Cooperative coevolution resolves the shortcomings of differential evolution to some extent. In this paper, the grouping method used in the CC framework is the Extended Differential Grouping (XDG) algorithm, which excels on detecting interacting variables and grouping them. Through experimental analysis, it is found that the performance of CCDE is better when the variables are separable (variables are not related), and the performance of COVDE on most problems is better when the variables are not separable (variables are related).

Since the correlation of variables is not clear in reality, we propose an adaptive mechanism based on cooperative coevolution and covariance. The mechanism is divided into two parts, including two evolutionary methods. One method is to add XDG into the CC framework, and then use differential evolution to optimize each subcomponent; the other one is to optimize the population with differential evolution, and then use covariance to analyze the characteristics of the resulting population. The better evolutionary direction is determined at the beginning of each generation by adaptive selection based on the success rate of both methods, and the better evolutionary direction of each generation is obtained by adaptive selection based on the fitness of the population.

Most of those existing adaptive DE variants are intended to update operators and control parameter settings. Instead, this paper proposed an adaptive mechanism with cooperative coevolution and covariance for differential evolution proposed which chooses between two evolutionary methods according to the fitness during the evolution so that CCDE can solve problems with separable variables and COVDE can solve problems with non-separable variables.

An adaptive mechanism with cooperative coevolution and covariance for differential evolution proposed in this paper can effectively solve optimization problems with unknown separability. Compared with COVDE, CCDE, DE, ADLADE, CSDE and TVDE, A-CC/COV-DE overperforms

the results of its counterparts, showing a competitive advantage when solving the optimization problems in the CEC2014 benchmark suite. However, the algorithm we proposed still has some deficiencies in some aspects, which will be improved in subsequent research.

## REFERENCES

- [1] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995.
- [2] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [3] G. Sun, Y. Lan, and R. Zhao, "Differential evolution with Gaussian mutation and dynamic parameter adjustment," *Soft Comput.*, vol. 23, no. 5, pp. 1615–1642, Mar. 2019.
- [4] G. Sun, Y. Wu, L. Deng, and K. Wang, "Elite representative based individual adaptive regeneration framework for differential evolution," *IEEE Access*, vol. 8, pp. 61226–61245, 2020.
- [5] G. Sun, G. Xu, and N. Jiang, "A simple differential evolution with time-varying strategy for continuous optimization," *Soft Comput.*, vol. 24, no. 4, pp. 2727–2747, Feb. 2020.
- [6] L. Deng, C. Li, and G. Sun, "An adaptive dimension level adjustment framework for differential evolution," *Knowl.-Based Syst.*, vol. 206, no. 4, pp. 1–20, 2020.
- [7] L.-B. Deng, L.-L. Zhang, N. Fu, H.-L. Sun, and L.-Y. Qiao, "ERG-DE: An elites regeneration framework for differential evolution," *Inf. Sci.*, vol. 539, pp. 81–103, Oct. 2020.
- [8] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
- [9] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, Kraków, Poland, vol. 6239, 2011, pp. 300–309.
- [10] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft Comput.*, vol. 18, pp. 232–247, May 2014.
- [11] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.
- [12] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
- [13] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.
- [14] Y. Sun, M. Kirley, and S. K. Halgamuge, "Extended differential grouping for large scale global optimization with direct and indirect variable interactions," in *Proc. Annu. Conf. Genetic Evol. Comput.*, Madrid, Spain, Jul. 2015, pp. 313–320.
- [15] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [16] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.
- [17] G. A. Trunfio, P. Topa, and J. Wąs, "A new algorithm for adapting the configuration of subcomponents in large-scale optimization with cooperative coevolution," *Inf. Sci.*, vol. 372, pp. 773–795, Dec. 2016.
- [18] Z. Ren, A. Chen, M. Wang, Y. Yang, Y. Liang, and K. Shang, "Bi-hierarchical cooperative coevolution for large scale global optimization," *IEEE Access*, vol. 8, pp. 41913–41928, 2020.
- [19] G. A. Trunfio, "Adaptation in cooperative coevolutionary optimization," in *Adaptation and Hybridization in Computational Intelligence*. Cham, Switzerland: Springer, 2015, pp. 91–109.
- [20] M. A. Meselhi, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Contribution based co-evolutionary algorithm for large-scale optimization problems," *IEEE Access*, vol. 8, pp. 203369–203381, 2020.
- [21] Q. Yang, W.-N. Chen, and J. Zhang, "Evolution consistency based decomposition for cooperative coevolution," *IEEE Access*, vol. 6, no. 1, pp. 51084–51097, 2018.

- [22] X.-M. Hu, F.-L. He, W.-N. Chen, and J. Zhang, "Cooperation coevolution with fast interdependency identification for large scale optimization," *Inf. Sci.*, vol. 381, pp. 142–160, Mar. 2017.
- [23] Y.-H. Jia, Y.-R. Zhou, Y. Lin, W.-J. Yu, Y. Gao, and L. Lu, "A distributed cooperative co-evolutionary CMA evolution strategy for global optimization of large-scale overlapping problems," *IEEE Access*, vol. 7, pp. 19821–19834, 2019.
- [24] J. Shao, D. L. Luo, Y. Xu, and H. B. Duan, "Cooperative path planning for multiple robots with motion constraints in obstacle-strewn environment," *IEEE Access*, vol. 7, pp. 132286–132301, 2019.
- [25] D. Luo, J. Shao, Y. Xu, Y. You, and H. Duan, "Coevolution pigeon-inspired optimization with cooperation-competition mechanism for multi-UAV cooperative region search," *Appl. Sci.*, vol. 9, no. 5, pp. 827–846, Feb. 2019.
- [26] F. Wei, Y. Wang, and T. Zong, "A novel cooperative coevolution for large scale global optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, San Diego, CA, USA, Oct. 2014, pp. 738–741.
- [27] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, Oct. 1994, pp. 249–257.
- [28] M. N. Omidvar, Y. Mei, and X. Li, "Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, Jul. 2014, pp. 1305–1312.
- [29] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Proc. Congr. Evol. Comput.*, vol. 2, May 2001, pp. 1101–1108.
- [30] M. N. Omidvar, X. Li, Z. Yang, and X. Yao, "Cooperative co-evolution for large scale optimization through more frequent random grouping," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1754–1761.
- [31] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2010, pp. 300–309.
- [32] M. N. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.
- [33] J. Yang, Q. Jiang, and L. Wang, "An adaptive encoding learning for artificial bee colony algorithms," *J. Comput. Sci.*, vol. 30, pp. 11–27, Jan. 2019.



**JUAN DIEGO PRADO** received the B.S. degree in computer engineering from Universidad del Valle, Cali, Colombia, in 2017. He is currently pursuing the M.S. degree in computer application technology with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include evolutionary algorithms, machine learning, artificial intelligence, and software development.



**WENJUAN HE** received the B.S. degree in computer management and application from the Xi'an University of Finance and Economics, Xi'an, China, in 1997, and the M.S. degree in computer science and technology from the Xi'an University of Technology, Xi'an, in 2004. She is currently working with the Shaanxi Key Laboratory for Network Computing and Security Technology. Her current research interests include evolutionary algorithms and artificial intelligence.



**HAIYAN JIN** received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2005 and 2007, respectively. She is currently a Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an. Her current research interests include computer vision, image processing, intelligent information processing, and intelligent optimization algorithms.



**QIAOYONG JIANG** received the B.S. degree in applied mathematics from Wenzhou University, Wenzhou, China, in 2008, the M.S. degree in computational mathematics from the Beifang University of Nationalities, Yinchuan, China, in 2011, and the Ph.D. degree in pattern recognition and intelligent systems from the Xi'an University of Technology, Xi'an, China. He is currently a Lecturer with the Faculty of Computer Science and Engineering, Xi'an University of Technology. His

current research interests include meta-heuristics, evolutionary computation, and multi-objective optimization.



**XIAOFAN WANG** received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2003, and the Ph.D. degree in computer science and engineering from Xidian University, Xi'an, in 2012. He is currently an Associate Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology. His current research interests include data mining, machine learning, and intelligent information processing.

...



**BIN WANG** received the B.S. degree in measurement and control technology and instrumentation from the Xi'an University of Technology, Xi'an, China, in 1994, and the M.S. degree in computer science and technology from Xidian University, Xi'an, in 2004. He is currently an Associate Professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology. His current research interests include evolutionary algorithms and artificial intelligence.



**LU REN** received the B.S. degree in computer science and technology from Yan'an University, Xi'an, China, in 2015. She is currently pursuing the M.S. degree in computer technology with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an. Her current research interests include evolutionary computing and deep learning.