# An Efficient Differential Evolution Via Both Top Collective and p-Best Information

Nengxian Liu[1,2], Jeng-Shyang Pan[1,2,3], Jinfeng Lai[4], Shu-Chuan Chu[3], Trong-The Nguyen[2]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] College of Mathematics and Computer Science, Fuzhou University, China
[3] Fujian Provincial Key Lab of Big Data Mining and Applications, Fujian University of Technology, China
[4] School of Electronic Engineering, University of Electronic Science and Technology of China, China
lylnx@fzu.edu.cn, jspan@cc.kuas.edu.tw, lcf2018@uestc.edu.cn, scchu0803@gmail.com, vnthe@hpu.edu.vn

## Abstract

This study proposes a novel DE variant for global optimizations based on both top collective information and p-best information (called CIpBDE). A combined mutation strategy (CIpBM) takes advantage of the mutation strategies "target-to-ci_pbest/1" and "target-to-pbest/1" is introduced trying to escape from stuck of local optima. A modified crossover operation (CIpBX) is proposed to handle the stagnation of DE. CIpBX adopts a collective vector or top p-best individual based on probability to execute crossover operation when stagnation occurs. An improved parameter adaptation strategy is figured out to adaptability to adjust the parameters crossover probability and scale factor value in each generation. To evaluate the performance of CIpBDE, comprehensive experiments are conducted on the CEC2013 benchmark test suit with 28 functions. Experimental results show that CIpBDE outperforms the seven state-of-the-art DE variants. What's more, we also apply CIpBDE to the feature selection problem. Compared results on several standard data sets indicate that CIpBDE outperforms the four comparing algorithms in terms of classification accuracy.

**Keywords:** Differential evolution, Collective information, p-best information, Global optimization, Feature selection

## 1 Introduction

Different types of optimization problems exist in many real-world applications. Besides traditional deterministic optimization methods, many intelligent stochastic optimization algorithms have been designed to deal with these optimization problems. These algorithms are commonly simple, powerful and easy to implement. Some representative algorithms are genetic algorithm (GA) [1], particle swarm optimization algorithm (PSO) [2-3], artificial bee colony algorithm (ABC) [4-5], Cat Swarm Optimization (CSO) [6-7], differential evolution algorithm (DE) [8-11], Bat Algorithm [12], etc. Among these algorithms, the DE is a simple yet effective intelligent optimization algorithm, which was first introduced by Storn and Price [8] in 1995 for solving continuous optimization problems. Over the last two decades, DE has attracted the attention of many researchers in the fields of science and engineering, and it has been widely applied to solve various real-world problems [13-15].

The basic DE algorithm originates from the Genetic Annealing Algorithm [16] that combined the GA and the simulated annealing algorithm (SA) [17]. Hence, the three main evolutionary operators, selection, mutation and crossover adopted in GA are inherited into DE. However, the order of these operations in DE is mutation, crossover, and selection which is different from the operation sequence of GA [18-20]. Compared with other evolutionary algorithms, DE has been proven its competitive performance, but its performance is affected by the mutation operation, the crossover operation, and associated control parameters such as scale factor F, crossover rate $CR$ and population size $NP$. To improve the DE's performance, many researchers first make their research focus on developing new mutation operators [21-23], crossover operators [24-26] and parameter control approaches [9, 27-28]. Now, there are more than six mutation strategies (i.e., DE/best/1, DE/rand/1, DE/best/2, DE/rand/2, DE/target-to-best/1, DE/target-to-rand/1 and DE/target-to-pbest/1) and two crossover operators (i.e., binomial/uniform crossover and exponential crossover) in the DE family. These mutations and crossover operations are suitable for different problems. It is a challenging work to design a new mutation strategy with good exploration and exploitation ability. In [22], Zheng et al. proposed a novel powerful mutation strategy "target-to-ci_mbest/1" using collective information of several good individuals, but this mutation strategy may be not effective to target vector with poor fitness value.

*Corresponding Author: Jeng-Shyang Pan; E-mail: jspan@cc.kuas.edu.tw
DOI: 10.3966/160792642020052103002

Meanwhile, different settings of the control parameters have different characteristics [29]. For example, DE with a smaller *CR* value pays attention to local exploitation because the target vector will propagate more components to the trial vector. In contrast, DE with a higher *CR* value will lead to higher population diversity, because the trial vector can get more components from the donor/mutant vector. Similarly, DE with a smaller $F$ value pays more attention to local search ability. In contrast, DE with a larger $F$ value pays more attention to global search ability. As many works of literature report the claims and counter-claims for the selection of appropriate control parameters [13, 30-31], these may confuse scientific researchers and engineers who use DE. In fact, it is still an open problem to choose the appropriate parameter settings to get a good trade-off the DE's exploration and exploitation capacity.

On the other hand, DE has the problem of stagnation and premature convergence [32-34]. Stagnation means that the algorithm no longer generates better candidate solutions even though the whole population has not yet converged. While premature convergence means that the algorithm cannot generate better candidate solutions because the whole population has converged to the local optimal solution.

In this study, we introduce a novel DE, referred to as CIpBDE, which consists of three main components. Firstly, we propose a combined mutation strategy CIpBM which takes the advantages of the mutation strategies "target-to-ci_pbest/1" and "target-to-pbest/1". The mutation strategy "target-to-ci_pbest/1" is a variant of the "target-to-ci_mbest/1". The "ci_pbest" means collecting information from the top $p*NP$ best individuals of the population. The letter p is a linear decreased value which is different from the letter m in "ci_mbest", where m is a random integer and, $m \in [1,i]$, governed by the rank $i$ target individual [22]. The mutation strategy "target-to-pbest/1" was introduced in the literature [21] for JADE. Second, either collective vector, which is generated by weighted contributions from top $p*NP$ best individuals, or the top p-best individual based on probability is used to perform crossover operation when stagnation occurs. This crossover operation is beneficial to alleviate the problem of stagnation. Finally, a modified parameter adaptation strategy is used to adjust the parameters $CR$ and $F$ values in each generation with the help of successful parameter values which can produce better offspring in the last generation. We verify the proposed CIpBDE algorithm by using the CEC2013 benchmark test suit and a real-world problem of feature selection in machine learning. The experimental results indicate that CIpBDE performs better than the comparing algorithms.

The remainder of this paper is organized as follows. We briefly introduce the standard DE in section 2. Section 3 describes the related work of DE. Then, our enhancing DE using top collective information and p-Best information (CIpBDE) is presented in Section 4. The experimental results under CEC2013 benchmark set are presented in Section 5. Moreover, the application of CIpBDE to feature selection problem is provided in Section 6. Section 7 provides the conclusion of this paper.

## 2 Differential Evolution

DE is a robust population-based algorithm. In general, there are four essential components in DE, i.e., initialization, mutation, crossover or recombination, and selection. First, DE initials the population, and then DE gets into a cycle of evolutionary operations consisted of mutation, crossover or recombination and selection until a stop condition is satisfied. The detailed description of these four components is provided below.

**Initialization.** DE begins with an initial population, which consists of *NP* D-dimensional individuals $X_{i,g} = (\mathrm{x}_{i,1,g}, \mathrm{x}_{i,2,g}, ..., \mathrm{x}_{i,\mathrm{D},g})$, $i = 1,2,...\mathrm{NP}$, where *NP* is the size of the population, $D$ is the dimension of the optimization problem, and $g$ is the generation number, $g = 1,2,...,G_{\max}$. Each individual denotes a candidate solution to the problem and is initialized by the following equation:

$$x_{i,j,0} = x_{\min,j} + rand(0,1) \cdot (x_{\max,j} - x_{\min,j}) \; j = 1,2,...,D \quad \textbf{(1)}$$

where $rand(0,1)$ is a uniformly distributed random generator in the interval [0, 1], $j$ is the parameter index in the $i$th individual at the generation $g = 0$ and $X_{\min} = \{\mathrm{x}_{\min,1}, \mathrm{x}_{\min,2}, ..\mathrm{x}_{\min,\mathrm{D}}\}$, $X_{\max} = \{\mathrm{x}_{\max,1}, \mathrm{x}_{\max,2}, ..\mathrm{x}_{\max,\mathrm{D}}\}$ are the lower and the upper bounds of the variable $x_{i,j}$.

**Mutation.** After initialization, a mutation operation is utilized to create a mutation/donor vector $V_{i,g} = (v_{i,1,g}, v_{i,2,g}, ..., v_{i,\mathrm{D},g})$ for each individual (target vector) in the population. Six basic and widely used mutation strategies are expressed below.

DE/best/1: $$V_{i,g} = X_{best,g} + F \cdot (X_{\mathrm{r}1,g} - X_{\mathrm{r}2,g}) \quad \textbf{(2)}$$

DE/rand/1: $$V_{i,g} = X_{r1,g} + F \cdot (X_{\mathrm{r}2,g} - X_{\mathrm{r}3,g}) \quad \textbf{(3)}$$

DE/best/2: $$V_{i,g} = X_{best,g} + F \cdot (X_{\mathrm{r}1,g} - X_{\mathrm{r}2,g}) + F \cdot (X_{\mathrm{r}3,g} - X_{\mathrm{r}4,g}) \quad \textbf{(4)}$$

DE/rand/2: $$V_{i,g} = X_{r1,g} + F \cdot (X_{\mathrm{r}2,g} - X_{\mathrm{r}3,g}) + F \cdot (X_{\mathrm{r}4,g} - X_{\mathrm{r}5,g}) \quad \textbf{(5)}$$

DE/target-to-rand/1: $$V_{i,g} = X_{\mathrm{i},g} + F \cdot (X_{r1,g} - X_{\mathrm{i},g}) + F \cdot (X_{\mathrm{r}2,g} - X_{\mathrm{r}3,g}) \quad \textbf{(6)}$$

DE/ target-to-best/1: $$V_{i,g} = X_{\mathrm{i},g} + F \cdot (X_{\mathrm{best},g} - X_{\mathrm{i},g}) + F \cdot (X_{\mathrm{r}1,g} - X_{\mathrm{r}2,g}) \quad \textbf{(7)}$$

where $X_{\text{best},g}$ is the current best individual found so far, $g$ denotes the generation, the indices $r1-r5$ are the random and mutually different integers generated from the interval $[1,\text{NP}]$ and these five integers are also different from $i$. The positive scaling factor $F$ is used for scaling the difference vectors. Generally, a convention DE/x/y/z [14] is used for naming these different mutation strategies, where x represents the base vector of the mutant operation, y represents the number of pairs of difference vectors used for perturbing the base vector x, and z represents the scheme of crossover being used.

**Crossover.** After mutation, a crossover operation is utilized to create a trial vector $U_{i,g}=(\text{u}_{i,1,g},\text{u}_{i,2,g},...,\text{u}_{i,\text{D},g})$ for each individual by combining the components of the mutation and target vectors. Two basic crossover operations are uniform (or binomial) crossover and exponential (or two-point modulo) crossover in the DE family [13]. Usually, the widely used uniform crossover is mathematically described as.

$$u_{i,j,g}=\begin{cases} v_{i,j,g} & if\ (rand_{i,j}(0,1)\le CR\ or\ j=j_{rand}) \\ x_{i,j,g} & otherwise \end{cases} \quad \textbf{(8)}$$

where $rand_{i,j}(0,1)$ is a uniform random number in the interval $[0,1]$, the parameter $CR$ is a user-defined crossover rate in the interval $(0,1]$, and $j_{rand}$ is a uniform random integer in the range $[1,\text{D}]$ which guarantees that trial vector $U_{i,g}$ has at least one variable inherited from donor vector $V_{i,g}$.

**Selection.** After crossover, DE calculates the fitness value of the trial vector with the objective function of optimization problem. Then, the selection operation of DE uses a one-to-one competition to pick out the target vector or the trial vector to the next generation according to their fitness values. The selection operation for a minimization problem is described as.

$$X_{i,g+1}=\begin{cases} U_{i,g} & if\ f(U_{i,g})\le f(X_{i,g}) \\ X_{i,g} & otherwise \end{cases} \quad \textbf{(9)}$$

where $f(X_{i,g})$ and $f(U_{i,g})$ are the fitness values of the target vector $X_{i,g}$ and the trial vector $U_{i,g}$, respectively. Note that if a trial vector $U_{i,g}$ survives to the next generation, it is called a successful update; otherwise it is called an unsuccessful upgrade. Their corresponding parameters, including the crossover rate $CR$ and the scale factor $F$, are called successful parameters or unsuccessful parameters.

## 3 Related Works

Since the first DE algorithm was proposed in 1995, many improved DE algorithms have been proposed in the literature to improve its performance. In this part, we briefly review these related works from four aspects.

### 3.1 Improving Mutation and Crossover Operations in DE

Fan et al. presented a trigonometric mutation operation and embedded it into an original DE to propose a novel DE named TDE [35]. In this method, a new parameter $M_t$ is adopted to combine the trigonometric mutation operation with the "DE/rand/1" mutation operation of DE. DE algorithm with this modification can make a better trade-off between the robustness and the convergence speed. The authors use two well-known functions and two neural network training problems to demonstrate the performance of TDE. Mohamed presented a new trigonometric mutation operation, which is different from the Fan's trigonometric mutation operation in two aspects, i.e., the weight and the difference vector [36]. In this method, the trigonometric mutation operation also combines with the "DE/rand/1" mutation strategy based on a non-linear decreasing probability. A restart mechanism is also presented in this method to prevent DE from getting into premature convergence. The authors tested the proposed algorithm on a set of well-known unconstrained problems and indicated its advantages over some DE variants. Das et al. introduced a neighborhood-based mutation operation which is an improved form of the mutation scheme "target-to-best/1" [37]. It consists of local neighborhood-based mutation and global neighborhood-based mutation. This method combines the two trial vectors generated by local and global neighborhood mutation to generate the actual trial vector. The purpose of the neighborhood-based mutation operation is to make a trade-off the exploration and exploitation abilities of DE. Zhang et al. designed a novel DE mutation operation with or without the optional external archive [21], called "target-to-pbest/1", which is a generalized version of the "target-to-best/1". This mutation makes full use of useful information of the top best $p*NP$ individuals of the current population. This mutation with the optional external archive also explores the useful information of inferior individuals. Islam et al. proposed a novel DE mutation operation, named "target-to-gr_best/1", which is also a variant of the "target-to-best" [24]. The mutation utilizes useful information of the best individuals of a set of randomly selected individuals from the current population. Yu et al. presented a new mutation operation called "DE/lbest/1", which is a variant of the greedy mutation strategy DE/best/1 [38]. This method uses multiple local best individuals to replace the global best individual to guide the individual's evolution. This mutation operation makes a good trade-off between population diversity and convergence speed. Cui et al. proposed three novel mutation strategies for multiple subpopulations and each strategy is responsible for either exploration or exploitation [23]. Zheng et al. [13] designed novel mutation operation named "target-to-ci_mbest/1" which uses the collective information of the top m best individuals instead of the single best individuals or the top p best individual. The useful information of these m best individuals is weight combined being a part of the difference vector for mutation operation. The authors used the CEC2013 benchmark set to assess the efficiency and effectiveness of the proposed mutation strategy.

Islam et al. presented a novel p-Best crossover [24]. This crossover is a usual uniform crossover incorporated with a biased parent selection. In this crossover, a mutant vector exchanges its variables with one of the top $p*NP$ best individuals in population instead of the target individual having the same index. Wang et al. adopted the orthogonal design to proposed an orthogonal crossover operation [39]. This approach can search systematically and rationally in an area defined by the mutant vector and target vector. The goal of this orthogonal crossover is to alleviate the drawbacks of two commonly used crossovers binomial and exponential which can only yield a trial vector in a hyper-rectangle determined by the mutant vector and target vector. Wang et al. applied covariance matrix learning to build an appropriate coordinate system for crossover operation based on the current population distribution [40]. This crossover operation reduces the DE's dependence on the coordinate system to some extent and enhances DE's capability to solve high-variable correlation optimization problems. Guo et al. introduced an eigenvector-based crossover operation [41]. This method uses the eigenvectors information of the individuals' covariance matrix to make the crossover rotationally invariant. To keep the population diversity, the offspring can be randomly born from the parents with either the rotated coordinate system or the standard coordinate system. The experimental results demonstrated that this crossover operation significantly improves the performance of DE. Meng et al. designed an automatically generated matrix to implement the crossover operation [26]. The authors demonstrated the proposed matrix performed well on the benchmark functions.

## 3.2 Adapting the Control Parameter Setting in DE

Liu et al. adopted a fuzzy logic controller to propose a fuzzy adaptive DE (FADE) [41], which adjusts the control parameters both for crossover and mutation operations. The experimental results indicated that FADE had better performance than the classic DE on high dimensionality problems. Brest et al. presented a DE variant, called jDE, with self-adaptive parameter control [27]. In this method, the scale factor $F$ and crossover rate $CR$ are encoded into the individuals and adaptively updated in the evolution process. The initial control parameter values are $F_i = 0.5$ and $CR_i = 0.9$. During the evolution, $F_i$ and $CR_i$ are updated obeying the uniform random distributions $U(0.1, 0.9)$ and $U(0,1)$ with user pre-defined probabilities $\tau_1$ and $\tau_2$. The authors used 21 test functions to assess the performance of the jDE. Zhang et al. proposed an adaptive DE with optional external archive (JADE) [21]. In JADE, for each target individual, the scale factor $F_i$ and crossover rate $CR_i$ are adaptively updated from the Cauchy distribution $randc(\mu F, 0.1)$ and normal distribution $randn(\mu CR, 0.1)$ respectively. The $\mu F$ and $\mu CR$ are updated based on their previous values and the successful parameters $F$ and $CR$ values. The parameters $F$ and $CR$ of some DE variants MDE-pBX [24], SHADE [42], and L-SHADE [43] are also updated from the Cauchy distribution and normal distribution respectively. Yu et al. proposed an effective two-level adaptive parameter setting scheme for DE [38]. In the first population-level, the parameters $F_P$ and $CR_F$ are adaptively controlled based on the optimization states for the whole population. In the second individual-level, the parameters $F_i$ and $CR_i$ for every individual are adaptively generated based on the individual's fitness value and its distance with the global best individual. The authors used 33 test functions to evaluate the efficiency and effectiveness of the proposed ADE. Draa et al. presented a DE variant Sinusoidal DE (SinDE) [44]. This variant employs sinusoidal formulas to tune parameters scaling factor $F$ and crossover rate $CR$ values. The experimental results have shown the superiority of the SinDE especially for composition and multimodal functions. Meng et al. proposed a DE variant PALM-DE whose parameters with adaptive learning mechanism to prevent control parameters mis-interaction in some DE variants' evolution process [31]. In PALM-DE, the scale factor $F$ and crossover rate $CR$ of all individuals' are separated into different groups to achieve better optimization performance.

## 3.3 Ensemble of Trial Vector Generation Strategies and Control Parameter Settings

Qin et al. presented a self-adaptive DE (SaDE) algorithm [29]. In this method, both mutation strategies and the control parameter settings are self-adaptive updated based on the previous search information. SaDE adopts a strategy candidate pool with four trial vector generation strategies. For each generation, each individual chose one strategy according to selection probability. Moreover, each individual in SaDE will be assigned different control parameter values. The authors used a test suite of 26 functions to evaluate SaDE's performance. Wang et al. presented a composite DE algorithm called CoDE [45]. CoDE employs three mutation strategies and three parameter control schemes to create the trial vectors in a random manner. This algorithm is based on some useful experiences, obtained by the previous DE researchers, on choosing strategies and control parameters. The authors used CEC2005 benchmark functions to assess CoDE's performance. Mallipeddi et al. presented a DE variant, called EPSDE, with the ensemble of mutation schemes and parameters [46]. In this approach, EPSDE uses a strategy pool containing four distinct mutation schemes and a pool of control parameter values. In the process of evolution, a mutation strategy and a control parameter setting are selected according to their successful experience in the past generations to create a trial vector. Therefore, a successful combination of mutation strategy and parameter settings has a higher probability to create the trial vector. The proposed EPSDE is verified on a suite of bound-constrained problems. Mallipeddi presented a novel parameter adaptation approach for DE using an ensemble method and harmony search (HS) [47]. This algorithm can be considered as a variant of EPSDE. In this algorithm, the values of ensemble parameters $F$ and $CR$ are evolved through the optimization process of the HS algorithm. The authors used CEC2005 benchmark functions to evaluate this adaptation method. Wu et al. presented a multi-population based method to implement an adapted ensemble of three mutation strategies into a new variant of DE, called MPEDE [48]. In MPEDE, the entire population is dynamically divided into three indicator subpopulations with a relatively smaller size and one reward subpopulation with a relatively larger size. Each indicator subpopulation has a constituent mutation strategy and the reward subpopulation is allocated to the mutation strategy with current best performance as an extra reward. Therefore, better mutation strategies can adaptively get more computational resources during the evolution process. Awad et al. presented a new ensemble parameters DE variant name EsDEr-NR which combines a Cauchy distribution and two sinusoidal formulas to tune the control parameters [28]. EsDEr-NR also uses a restart method at later stage to improve the quality of the obtained solutions. Besides, a novel approach niching-based reduction scheme is used to adapt the population size. The authors used two CEC test suites to verify the performance of the EsDEr-NR.

## 3.4 Stagnation of DE

As described in the introduction section, DE suffers from the problems of stagnation and premature convergence. Stagnation is the status that the algorithm is incapable of generating better candidate solutions even if the population maintains a certain degree of diversity. In this study, the stagnant individual is identified by the consecutive unsuccessful update $CUU_i (\mathrm{i}=1,2,...,\mathrm{NP})$, and it can be defined as follows.

$$CUU_{i,g+1} = \begin{cases} 0 & if\ \ f(U_{i,g}) \le f(X_{i,g}) \\ CUU_{i+g} + 1 & otherwise \end{cases} \quad \textbf{(10)}$$

where $CUU_{i,g}$ represents the consecutive unsuccessful update for the $i$th individual of the population, and its initial value is set as 0. If $CUU_{i,g} > T$, where $T$ denotes the user-defined threshold of stagnation, it indicates $i$th individual is in stagnation [22, 34, 49]. This study $T$ is set to 90, according to the literature [22].

To address the stagnation and premature convergence problems, some effective approaches have been proposed. Guo et al. proposed a DE variant, called SPS-DE, which solved the stagnation problem of DE by employing a successful parent selection framework [49]. In this approach, when an individual falls into stagnation, the current successful individuals which are stored in the archive will be chosen as the parent to produce offspring. Zhou et al. proposed a guiding archiving framework to help DE escape from the stagnation status. (GAR-DE) [50]. In this approach, some high-quality individuals are saved in a guiding archive during evolution. If an individual drops into stagnation, GAR-DE selects another individual from the guiding archive to take the place of the base vector in the mutation operation to guide the evolution. Cui et al. solved the stagnation and premature convergence issues with a novel shift mechanism (SM) [23]. In SM, if stagnation occurs, some inferior individuals will be shifted to a neighborhood of one of the top $p \cdot NP$ solutions to promote convergence. If premature convergence occurs, some inferior individuals will be unconditionally shifted to random positions to improve the diversity of the population. Cui et al. proposed Tracking Mechanism (TM) along with Backtracking Mechanism (BTM) framework to deal with the stagnation and premature convergence problems [34]. This approach uses TM to promote convergence if the population drops into stagnation. This approach uses BTM to improve the population diversity if the population drops into premature convergence. Zheng et al. proposed a variant of DE powered by collective information (CPIDE) [22]. In CPIDE, m best individuals are weighted combined to form a new collective vector which works as a part of the difference vector in mutation. When the population drops into stagnation, this collective vector takes part in the crossover operation to help DE escape from the stagnation status.

## 4 DE with Top Collective Information and p-Best Information (CIpBDE)

In this part, the main idea of CIpBDE which using top collective information and p-Best information is described in detail. First, we introduce the combined mutation strategy adopted in CIpBDE and then we describe the crossover operation utilized in CIpBDE. Finally, the new proposed parameters control scheme is discussed.

### 4.1 Combined Mutation Strategy in CIpBDE

Empirically, the optimization performance of the DE algorithm is influenced heavily by the mutation strategy. Generally, the purpose of mutation operation for DE is to control the search direction, and the different DE mutation operations have different characteristics. "DE/rand/1" and "DE/rand/2" are essentially random strategies. Both the base and difference vectors of these two strategies are chosen in a random way. Hence, their exploration ability is strong, but their convergence speed may be slow. "DE/best/1" and "DE/target-to-best/1" employ the reliable information of the best individual found so far. The base vector of these strategies is the current or the best vector, and the difference vector is constructed in a random way. Hence, these strategies have a good exploitative ability and have a fast convergence speed, but may lose their diversity and cause premature convergence. Due to some shortcomings of above mutation strategies, a very powerful mutation strategy "DE/target-to-pbest/1" with or without archive was proposed in [21], which is given below.

$$V_{i,g} = X_{i,g} + F \cdot (X_{pbest,g} - X_{i.g}) + F \cdot (X_{r1,g} - \overline{X_{r2.g}}) \quad \textbf{(11)}$$

where $X_{pbest,g}$ is randomly selected from the top $p*NP$ best individuals of the population, and $p$ is in the interval $(0,1]$. $\overline{X_{r2.g}}$ is randomly chosen from the union of the current population and the external archive. This mutation strategy has two advantages. (1) It uses one of the top $p*NP$ best individuals to trade off the greediness of the mutation and the convergence speed. (2) It can improve the population diversity by exploring inferior individuals in the archive, and the archive provides the information which denotes promising progress directions toward the global optimum.

Another powerful and effective mutation strategy, "DE/target-to-ci_mbest/1" using the collective information of $m$ top ranking individuals from the population is proposed [22], which is given below.

$$V_{i,g} = X_{i,g} + F \cdot (X_{ci\_mbest^i,g} - X_{i.g}) + F \cdot (X_{r1,g} - X_{r2g}) \quad \textbf{(12)}$$

where $X_{ci\_mbest^i,g}$ is called collective vector, which is a weighted combination of the top $m$ ( $m$ , $m \in [0,i]$ , is a randomly integer for $X_{i,g}$ ) best individuals of the population with fitness values equal to or better than $X_{i,g}$ . $X_{ci\_mbest^i,g}$ is calculated as fellows.

$$X_{ci\_mbest^i,g} = \sum_{k=1}^{m} w_k X_{k,g}, \; w_k = \frac{m-k+1}{\sum_{k=1}^{m} k}, \quad \textbf{(13)}$$
$$k = 1, 2, ..., m$$

where $w_k$ are the weighting factors which represent the contributions of different individuals. The experimental results of paper [22] demonstrate that it can balance exploitative and explorative search and it has better performance than mutation strategies "target-to-best/1", "target-to-pbest/1" and "target-to-gr_best/1". However, sometimes the calculated collective vector $X_{ci\_mbest^i,g}$ is not in a promising position, so it cannot guide the population to the global optimal position, as shown in Figure 1. From Figure 1, we can see that the collective vector $X_{ci\_mbest^i,g}$ is far away from the global optimal. In addition, sometimes the random $m$ will be large for the target vector with poor fitness value. In this case, this mutation strategy is not effective.

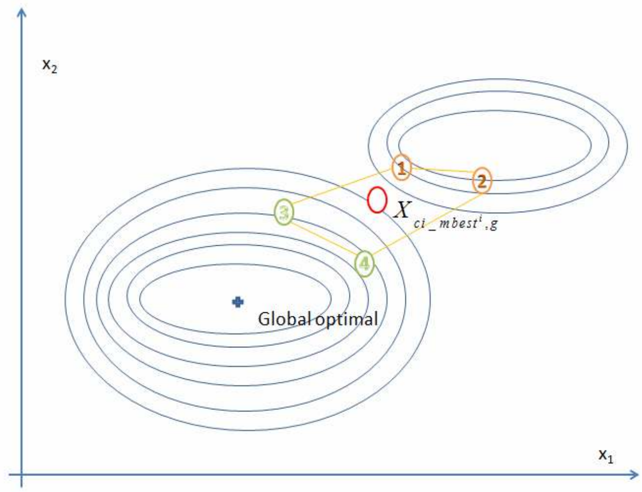


**Figure 1.** Illustration of the unpromising position of collective vector

To alleviate these shortcomings abovementioned, we design the following combined mutation strategy, called CIpBM, by employing two powerful mutation strategies "DE/target-to-ci_mbest/1" and "DE/target-to-pbest/1" with a predefined constant probability.

$$V_{i,g} = \begin{cases} X_{i,g} + F \cdot (X_{ci\_pbest,g} - X_{i.g}) + F \cdot (X_{r1,g} - \overline{X_{r2.g}}) \\ \text{if } rand < 0.5 \\ X_{i,g} + F \cdot (X_{pbest,g} - X_{i.g}) + F \cdot (X_{r1,g} - \overline{X_{r2.g}}) \\ \text{otherwise} \end{cases} \quad \textbf{(14)}$$

where $X_{ci\_pbest,g}$ is a collective vector which is similar to the $X_{ci\_mbest^i,g}$. $\overline{X_{r2,g}}$ is randomly chosen from the union of the external archive and the current population. Parameter p is independent of the target vector index $i$ which is a real number, $p \in (0,1]$. The value for p is linearly adjusted in the following way.

$$p = p_{\max} - (p_{\max} - p_{\min}) \cdot gen/(MaxGen) \quad \textbf{(15)}$$

where $gen$ denotes the current generation number, $gen = [1, 2, ..., MaxGen]$, $MaxGen$ is the maximum number of generations. $p_{\max}$ is the maximum value of $p$ and $p_{\min}$ is the minimum value of $p$. The reduction rule of $p$ is beneficial to exploration at the early stage of the evolution and exploitation for the later stage. This combined mutation strategy can take advantage of two original mutation strategies.

## 4.2 Crossover in CIpBDE

As described in section 3.4, DE has the problem of stagnation, which seriously affects the performance of DE. When stagnation is happening to DE, it means that some target individuals in the current population are no longer updated and their consecutive unsuccessful update index $CUU$ reaches the threshold $T$. To address this problem, Zheng et al. proposed a collective information-based crossover (CIX) [22]. When the $i$th target individual falls into stagnation, i.e., $CUU_{i,g} > T$, CIX is defined as follows:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand_1 \le CR \text{ or } j = j_{rand} \\ x_{ci\_mbest^i,j,g} & \text{otherwise} \end{cases} \quad \textbf{(16)}$$

The experiment results of paper [22] demonstrate that CIX is an efficient and effective crossover to address stagnation problem. However, when the calculated collective vector $X_{ci\_mbest^i,g}$ is not in the promising position as shown in Figure 1, it will be ineffective crossover operation. In this study, we incorporate the crossover operation with p-Best crossover [24], named the modified crossover operation CIpBX. When $CUU_{i,g} > T$, T is a threshold, CIpBX is defined as follows:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand_1 \le CR \text{ or } j = j_{rand} \\ x_{ci\_pbest,j,g} & \text{if } rand_2 < 0.5 \\ x_{pbest,j,g} & \text{otherwise} \end{cases} \quad \textbf{(17)}$$

where $x_{ci\_pbest,j,g}$ is the $j$th dimension of collective vector $X_{ci\_pbest,g}$, $x_{pbest,j,g}$ is the $j$th dimension of vector $X_{pbest,g}$. $rand_1$ and $rand_2$ are random numbers in the interval $[0,1]$.

## 4.3 Parameter Adaptation Approach in CIpBDE

The DE's performance is sensitively affected by parameters $F$ and $CR$ [38]. In literature [21-22, 27], some parameter adaption schemes utilized feedback information to solve this issue. To make the proposed CIpBDE insensitive to parameters $F$ and $CR$, a modified parameter adaption scheme based on literature [21] is used to tune the parameters $F$ and $CR$ which is described in the following.

At every generation, the $F_i$ for each individual $X_{i,g}$ is independently generated from a Cauchy distribution using location parameter $\mu F$ and scale parameter 0.1, as follows.

$$F_i = randc_i(\mu F, 0.1) \quad \textbf{(18)}$$

where $\mu F$ is initialized to 0.5. $F_i$ is truncated to 1.0 in the case of $F_i > 1.0$, while $F_i$ is regenerated using Eq. (18) in the case of $F_i < 0$. At every generation, all successful parameter $F_i$ is saved in the set $S_F$. At the end of every generation, if $S_F$ is not empty, $\mu F$ is updated according to the first Equation of Eq. (19). If $S_F$ is empty, it means that the current parameter $\mu F$ may be unsuitable at the current stage. Therefore, we update the $\mu F$ according to the second or third Equations of Eq. (19) with a user-defined constant probability.

$$\mu F = \begin{cases} (1-c) \cdot \mu F + c \cdot mean_L(S_F) & \text{if } S_F \ne \varnothing \\ (1-c) \cdot \mu F + c \cdot rand \cdot (1-\mu F) & \text{if } S_F = \varnothing \\ \text{and } rand < \tau_1 & \\ \mu F & \text{otherwise} \end{cases} \quad \textbf{(19)}$$

where the control parameter $c$ is the learning rate [21], the term $mean_L(\cdot)$ represents the Lehmer mean, the parameter $\tau_1$ is a user-defined constant probability. The parameter $\tau_1$ is set to 0.1 through experiment.

Similarly, at every generation, the $CR_i$ for every individual $X_{i,g}$ is independently generated from a Gaussian distribution with mean $\mu CR$ and standard deviation 0.1, as follows.

$$CR_i = randn_i(\mu CR, 0.1) \quad \textbf{(20)}$$

where $\mu CR$ is initialized to 0.5. $CR_i$ is truncated to 1.0 in the case of $CR_i > 1.0$, while $CR_i$ is truncated to 0 in the case of $CR_i < 0$. At every generation, all successful parameter $CR_i$ is saved in the set $S_{CR}$. At the end of every generation, if $S_{CR}$ is not empty, $\mu CR$ is updated according to the first Equation of Eq. (21). If $S_{CR}$ is empty, it means that the current parameter

$\mu CR$ may be unsuitable at the current stage. Therefore, we update the $\mu CR$ according to the second or third Equation of Eq. (21) with a user-defined constant probability.

$$\mu CR = \begin{cases} (1-c)\cdot \mu CR + c\cdot mean_A(S_{CR}) & if\ S_{CR} \neq \varnothing \\ (1-c)\cdot \mu CR + c\cdot rand\cdot(1-\mu CR) & if\ S_{CR} = \varnothing \\ and\ rand < \tau_2 & \\ \mu CR & otherwise \end{cases} \quad \textbf{(21)}$$

where the control parameter $c$ is the learning rate [21], the term $mean_A(\cdot)$ represents the usual arithmetic mean, the parameter $\tau_2$ is a user-defined constant probability. The parameter $\tau_2$ is set to 0.1 through experiment.

The pseudo code of the whole CIpBDE algorithm is given in Algorithm 1. Its flowchart is given in Figure 2.

**Algorithm 1.** Pseudo-code of the CIpBDE

1. **Initialization**: Randomly generate initial population $P$, set $Gen=1$, $FESmax=10000*D$, $FES=0$, $T=90$, $p_{\min}=0.1$, $p_{\max}=0.2$, $\tau_1=\tau_2=0.1$, $\mu F=0.5$, $\mu CR=0.5$, $c=0.1$, $CUU_{i=1:NP}=0$
2. Compute the fitness value of the population $P$; $FES=FES+NP$;
3. **while** $FES \le FES\max$ **do**
4. $S_F=\varnothing$, $S_{CR}=\varnothing$;
5. Sort the population $P$ according to the fitness
6. Update parameter $p$ using Eq (15)
7. **for** $i=1; i\le NP; i++$ **do**
8. Generate $F_i$ using Eq (18), $CR_i$ using Eq (20);
9. Generate mutant vector $V_{i,g}$ using Eq (14);
10. **if** $CUU_i < T$ **then**
11. Generate trial vector $U_{i,g}$ using Eq (8);
12. **else**
13. Generate trial vector $U_{i,g}$ using Eq (17);
14. **end if**
15. Compute the fitness value $f(U_i)$; $FES=FES+1$;
16. **if** $f(U_i)\le f(X_i)$ **then**
17. $X_{i,g+1}=U_{i,g}$; $CUU_i=0$; $F_i \to S_F$; $CR_i \to S_{CR}$;
18. **else**
19. $X_{i,g+1}=X_{i,g}$; $CUU_i=CUU_i+1$;
20. **end if**
21. **end for**
22. Update $\mu F$ using Eq. (19); Update $\mu CR$ using Eq.(21);
23. $Gen=Gen+1$;
24. **end while**

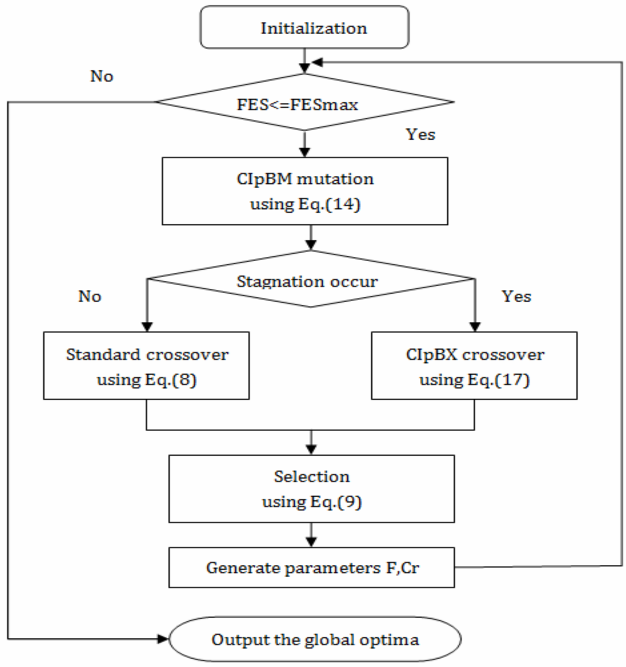


**Figure 2.** The flowchart of the proposed CIpBDE

## 5 Experimental Analysis

In this part, CIpBDE was assessed on 28 functions proposed for IEEE CEC2013 [51]. Among these 28 functions, the first five functions F1-F5 are unimodal, the following fifteen functions F6-F20 are basic multimodal, and the last eight functions F21-F28 are composition. All these benchmark functions are treated as black-box test problems, and the search ranges of them are confined to $[-100,100]^D$ (D is decision variables number). The detailed definitions of these 28 benchmark functions are given in [51]. All these benchmark functions are shifted to the same global optima $O=\{o_1,o_2,...,o_d\}^T$.

In the experiment, the number of dimension $D$ for all functions is set to 30. The maximal number of function evaluation is set as $D\times 10^4$ (NFE= $D\times 10^4$). The contrasted algorithms run 51 times independently on each function. Both the best and the mean/standard deviation (Std) of the function errors $\Delta f = f_i - f_i^*$ are reported. Symbols "-", "=" and "+" in parentheses behind the values represent "worse performance", "similar performance" and "better performance" than our approach respectively. The Wilcoxon signed-rank test with significance level $\alpha = 0.05$ is used for comparing the "Mean/Std". We use the rule "The smaller the better" to compare the "Best" values in form of arithmetic values. All the experiments were conducted on a computer with Intel(R) Core(TM) i5 3.3 GHz dual-core CPU and 8.0 GB of RAM in Operating System Windows 7. All the algorithms were implemented in Matlab 2016a.

The proposed CIpBDE is compared with several powerful state-of-the-art DE variants including CIPDE [22], CobiDE [40], CoDE [45], JADE [21], jDE [27], SaDE [29], and SHADE [42] on F1-F28 benchmark functions. These DE variants are chosen for their competitive performance and popularity. Table 1 gives all these contrasted algorithms' parameter settings according to the references. Table 2 and Table 3 report the best and the mean/standard deviation of the function errors respectively. Due to limited space, the comparisons of convergence speed for the contrasted algorithms are given in the supplementary file.

As shown in Table 2, our proposed CIpBDE has better performance than or at least comparable to the contrasted algorithms from the best value perspective of view. Our proposed CIpBDE algorithm exhibits either better or similar performance improvement in 20 out of 28 functions in comparing with CIPDE, improvement in 20 out of 28 functions in comparing with CobiDE, improvement in 25 out of 28 functions in comparing with CoDE, improvement in 23 out of 28 functions in comparing with JADE, improvement in 24 out of 28 functions in comparing with jDE,

**Table 1.** Recommended Parameter settings for all of these contrasted algorithms

| Algorithms | Parameters settings |
|---|---|
| CIPDE | $NP = 100, \mathrm{c} = 0.1, \mu \mathrm{F} = 0.7,$ $\mu \mathrm{CR} = 0.5, \mathrm{T} = 90$ |
| CobiDE | $NP = 100, \mathrm{pb} = 0.4, \mathrm{ps} = 0.5,$ $\mu F = 0.65\ or\ 1.0, \mu \mathrm{CR} = 0.1\ or\ 0.95$ |
| CoDE | $NP = 30, F = 1.0, \mathrm{CR} = 0.1, or\ F = 1.0,$ $\mathrm{CR} = 0.9\ or\ F = 0.8, \mathrm{CR} = 0.2$ |
| JADE | $NP = 100, \mathrm{c} = 0.1, \mu \mathrm{F} = 0.5,$ $\mu \mathrm{CR} = 0.5, p = 0.05$ |
| jDE | $NP = 100, CR = 0.9,$ $\tau_1 = \tau_2 = 0.1, \mathrm{F}_1 = 0.1, F_u = 0.9$ |
| SaDE | $NP = 50, F \sim N(0.5, 0.3), \mu CR = 0.5,$ $CR \sim N(\mu CR, 0.1), LG = 50$ |
| SHADE | $NP = 100, \mu \mathrm{F} = 0.5,$ $\mu \mathrm{CR} = 0.5, p = 0.2, \mathrm{H} = 100$ |
| CIpBDE | $NP = 100, \mathrm{c} = 0.1, \mu \mathrm{F} = 0.5, \mu \mathrm{CR} = 0.5,$ $\mathrm{p}_{\max} = 0.2, \mathrm{p}_{\min} = 0.1, \tau_1 = \tau_2 = 0.1, \mathrm{T} = 90$ |

**Table 2.** Comparison results of the best value of 51-run fitness error for CIpBDE with seven DE variants

| 30D | CIPDE | CobiDE | CoDE | JADE | jDE | SaDE | SHADE | CIpBDE |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00 |
| 2 | 4.0521E+02(+) | 9.2336E+01(+) | 2.9736E+04(-) | 6.4544E+02(-) | 2.0178E+04(-) | 6.0096E+03(-) | 3.6789E+02(+) | 5.3458E+02 |
| 3 | 1.5916E-12(+) | 2.7367E+01(-) | 6.7000E-06(-) | 9.0949E-13(+) | 6.7997E-05(-) | 2.6990E-07(-) | 9.0949E-13(+) | 2.0464E-12 |
| 4 | 1.9980E-06(-) | 5.3888E-11(+) | 1.7493E-03(-) | 1.2913E-09(+) | 2.6065E+00(-) | 4.3635E-01(-) | 5.7094E-10(+) | 1.5253E-08 |
| 5 | 0.0000E+00(=) | 2.2737E-13(-) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 1.1369E-13(-) | 0.0000E+00 |
| 6 | 1.1369E-13(=) | 1.2479E+00(-) | 1.5875E-06(-) | 1.1369E-13(=) | 9.5362E+00(-) | 9.1836E-08(-) | 0.0000E+00(+) | 1.1369E-13 |
| 7 | 6.1241E-02(+) | 1.9670E+00(-) | 2.1909E+00(-) | 1.6698E-01(-) | 3.3133E-01(-) | 7.2550E-01(-) | 2.7149E-01(-) | 9.1763E-02 |
| 8 | 2.0845E+01(-) | 2.0798E+01(+) | 2.0384E+01(+) | 2.0410E+01(+) | 2.0773E+01(+) | 2.0819E+01(-) | 2.0223E+01(+) | 2.0817E+01 |
| 9 | 1.3444E+01(-) | 4.6535E+00(+) | 8.1665E+00(+) | 2.0315E+01(-) | 9.4606E+00(-) | 8.3679E+00(+) | 2.3098E+01(-) | 9.1733E+00 |
| 10 | 9.8573E-03(-) | 5.6843E-14(+) | 7.3960E-03(=) | 0.0000E+00(+) | 7.3960E-03(=) | 2.7101E-02(-) | 0.0000E+00(+) | 7.3960E-03 |
| 11 | 0.0000E+00(=) | 1.8713E+00(-) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00(=) | 0.0000E+00 |
| 12 | 5.9698E+00(+) | 1.1940E+01(-) | 1.9899E+01(-) | 1.4413E+01(-) | 3.9706E+01(-) | 1.9899E+01(-) | 9.9893E+00(-) | 6.9647E+00 |
| 13 | 5.7106E+00(+) | 2.4202E+01(-) | 2.2088E+01(-) | 2.3824E+01(-) | 6.0785E+01(-) | 2.2939E+01(-) | 1.0491E+01(-) | 5.9755E+00 |
| 14 | 2.5013E-01(-) | 5.8417E+02(-) | 3.1251E-01(-) | 1.8190E-12(+) | 0.0000E+00(+) | 1.6088E+01(-) | 1.8190E-12(+) | 7.6098E-02 |
| 15 | 1.4961E+03(+) | 1.5984E+03(+) | 2.2583E+03(-) | 2.2913E+03(-) | 4.3157E+03(-) | 5.4454E+03(-) | 1.9267E+03(-) | 1.6186E+03 |
| 16 | 3.1495E-01(-) | 1.7165E-02(+) | 6.6695E-02(+) | 8.2394E-01(-) | 1.7730E+00(-) | 1.6740E+00(-) | 1.3400E-01(+) | 2.3525E-01 |
| 17 | 3.0467E+01(-) | 4.7724E+01(-) | 3.0434E+01(=) | 3.0434E+01(=) | 3.0434E+01(=) | 3.2747E+01(-) | 3.0434E+01(=) | 3.0434E+01 |
| 18 | 3.3732E+01(+) | 5.4198E+01(-) | 4.3145E+01(-) | 6.0021E+01(-) | 1.0107E+02(-) | 1.3118E+02(-) | 5.5312E+01(-) | 3.5539E+01 |
| 19 | 8.1537E-01(-) | 5.1213E+00(-) | 7.7061E-01(-) | 1.1453E+00(-) | 1.2914E+00(-) | 3.1550E+00(-) | 9.5468E-01(-) | 6.0195E-01 |
| 20 | 8.1967E+00(+) | 9.0441E+00(-) | 9.4752E+00(-) | 9.2622E+00(-) | 1.0737E+01(-) | 1.0153E+01(-) | 9.6223E+00(-) | 8.4302E+00 |
| 21 | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02(=) | 2.0000E+02 |
| 22 | 1.0610E+02(-) | 1.0945E+03(-) | 1.4737E+01(-) | 1.8720E+01(-) | 5.8446E+01(-) | 1.2510E+02(-) | 7.3716E+00(+) | 1.2420E+01 |
| 23 | 1.5038E+03(-) | 1.5159E+03(-) | 2.6974E+03(-) | 2.2767E+03(-) | 3.7809E+03(-) | 5.0174E+03(-) | 2.1034E+03(-) | 1.2788E+03 |
| 24 | 2.0127E+02(-) | 2.0260E+02(-) | 2.0434E+02(-) | 2.0061E+02(-) | 2.0007E+02(+) | 2.0457E+02(-) | 2.0083E+02(-) | 2.0013E+02 |
| 25 | 2.4268E+02(-) | 2.3644E+02(+) | 2.4114E+02(-) | 2.4236E+02(-) | 2.3573E+02(+) | 2.0054E+02(+) | 2.3383E+02(+) | 2.4003E+02 |
| 26 | 2.0000E+02(-) | 2.0000E+02(-) | 2.0000E+02(-) | 2.0000E+02(-) | 2.0000E+02(-) | 2.0000E+02(-) | 2.0000E+02(=) | 2.0000E+02 |
| 27 | 3.0999E+02(-) | 3.4275E+02(-) | 3.2663E+02(-) | 3.1107E+02(-) | 3.2123E+02(-) | 3.2658E+02(-) | 3.0372E+02(+) | 3.0540E+02 |
| 28 | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02(=) | 3.0000E+02 |
| -/=/+ | 14/6/8 | 17/3/8 | 18/7/5 | 16/7/5 | 17/7/4 | 21/5/2 | 11/6/11 | -/-/- |

**Table 3.** Comparison results of mean and Std of 51-run fitness error for CIpBDE with seven DE variants

| 30D | CIPDE | CobiDE | CoDE | JADE |
|---|---|---|---|---|
| | Mean/Std | Mean/Std | Mean/Std | Mean/Std |
| 1 | 0.0000E+00/0.0000E+00(=) | 0.0000E+00/0.0000E+00(=) | 0.0000E+00/0.0000E+00(=) | 0.0000E+00/0.0000E+00(=) |
| 2 | 9.3962E+03/7.1722E+03(-) | 1.3243E+04/1.054E+04(-) | 8.2127E+04/4.5164E+04(-) | 7.1972E+03/5.6487E+03(-) |
| 3 | 6.3986E+05/2.1860E+06(-) | 1.1023E+03/2.3348E+03(-) | 6.9114E+05/2.2153E+06(-) | 2.3499E+05/9.3590E+05(=) |
| 4 | 6.1405E+03/9.9447E+03(=) | 4.5506E-10/4.8110E-10(+) | 7.9173E-02/9.3842E-02(+) | 7.4965E+03/1.4972E+04(+) |
| 5 | 9.8083E-14/3.9511E-14(-) | 3.7673E-13/1.2555E-13(-) | 0.0000E+00/0.0000E+00(+) | 9.3624E-14/4.3771E-14(-) |
| 6 | 1.0356E+00/5.1769E+00(-) | 3.6465E+00/3.3738E+00(-) | 2.4244E+00/7.0751E+00(-) | 2.0712E+00/7.1703E+00(=) |
| 7 | 2.5448E+00/2.4410E+00(=) | 7.0694E+00/3.4651E+00(-) | 1.1649E+01/7.2934E+00(-) | 2.9711E+00/2.8873E+00(=) |
| 8 | 2.0947E+01/4.8720E-02(=) | 2.0952E+01/5.1050E-02(=) | 2.0740E+01/1.1904E-01(+) | 2.0915E+01/9.1676E-02(=) |
| 9 | 1.9363E+01/2.7813E+00(-) | 1.0182E+01/3.3669E+00(+) | 1.4752E+01/2.9993E+00(+) | 2.6761E+01/1.6755E+00(-) |
| 10 | 6.7910E-02/3.8206E-02(-) | 6.1830E-03/8.4526E-03(+) | 3.3951E-02/2.7969E-02(=) | 3.9991E-02/2.5227E-02(=) |
| 11 | 0.0000E+00/0.0000E+00(=) | 6.4187E+00/2.0904E+00(-) | 0.0000E+00/0.0000E+00(=) | 0.0000E+00/0.0000E+00(=) |
| 12 | 1.5822E+01/5.5869E+00(=) | 3.4693E+01/1.1780E+01(-) | 3.8647E+01/9.6328E+00(-) | 2.3129E+01/4.1387E+00(-) |
| 13 | 1.9491E+01/8.0306E+00(+) | 7.0330E+01/2.6487E+01(-) | 7.7105E+01/2.7113E+01(-) | 4.8141E+01/1.2033E+01(-) |
| 14 | 5.9795E-01/3.8020E-01(-) | 8.7368E+02/1.3476E+02(-) | 3.1456E+00/3.0310E+00(-) | 3.3474E-02/2.3948E-02(+) |
| 15 | 2.7119E+03/6.1588E+02(=) | 2.7427E+03/5.2300E+02(=) | 3.3391E+03/5.5662E+02(-) | 3.2668E+03/3.3296E+02(-) |
| 16 | 2.1522E+00/6.7961E-01(-) | 1.9092E+00/9.1874E-01(-) | 3.2570E-01/2.0054E-01(+) | 1.9073E+00/6.4273E-01(-) |
| 17 | 3.0518E+01/3.6992E-02(-) | 5.3818E+01/2.4424E+00(-) | 3.0440E+01/2.1304E-02(-) | 3.0434E+01/8.0389E-15(=) |
| 18 | 4.0500E+01/7.5021E+00(+) | 1.0602E+02/5.4637E+01(-) | 6.3991E+01/1.3373E+01(-) | 7.6421E+01/6.3556E+00(-) |
| 19 | 1.0603E+00/1.5678E-01(-) | 6.3504E+00/5.8271E-01(-) | 1.5968E+00/4.4499E-01(-) | 1.4762E+00/9.5593E-02(-) |
| 20 | 9.7413E+00/7.1374E-01(=) | 1.1447E+01/8.5903E-01(-) | 1.0678E+01/5.9092E-01(-) | 1.0456E+01/5.0898E-01(-) |
| 21 | 2.8858E+02/6.7287E+01(=) | 3.2919E+02/1.1676E+02(=) | 3.2457E+02/8.5823E+01(=) | 3.0180E+02/7.8138E+01(=) |
| 22 | 1.1436E+02/1.5046E+01(-) | 1.5104E+03/2.5912E+02(-) | 1.1594E+02/1.8021E+01(-) | 9.3145E+01/3.0505E+01(=) |
| 23 | 2.6142E+03/7.3241E+02(=) | 2.5974E+03/4.5888E+02(=) | 3.7686E+03/6.8355E+02(-) | 3.5193E+03/4.8098E+02(-) |
| 24 | 2.0728E+02/4.8723E+00(=) | 2.0459E+02/2.2509E+00(=) | 2.2198E+02/9.4856E+00(-) | 2.1101E+02/9.5399E+00(-) |
| 25 | 2.5954E+02/8.2620E+00(=) | 2.4980E+02/7.1605E+00(+) | 2.5377E+02/6.3474E+00(+) | 2.7306E+02/1.0436E+01(-) |
| 26 | 2.1246E+02/3.4403E+01(-) | 2.0000E+02/1.2235E-03(-) | 2.1046E+02/3.6248E+01(-) | 2.1644E+02/4.5559E+01(-) |
| 27 | 4.3595E+02/9.9656E+01(-) | 4.9396E+02/1.1482E+02(-) | 5.8483E+02/1.1247E+02(-) | 7.6165E+02/2.1918E+02(-) |
| 28 | 3.0000E+02/0.0000E+00(=) | 3.0000E+02/2.5404E-09(=) | 3.0000E+02/0.0000E+00(=) | 3.0000E+02/0.0000E+00(=) |
| -/=/+ | 13/13/2 | 17/7/4 | 17/5/6 | 15/11/2 |

| 30D | jDE | SaDE | SHADE | CIpBDE |
|---|---|---|---|---|
| | Mean/Std | Mean/Std | Mean/Std | Mean/Std |
| 1 | 4.4583E-15/3.1839E-14(=) | 0.0000E+00/0.0000E+00(=) | 4.0125E-14/8.7542E-14(-) | 0.0000E+00/0.0000E+00 |
| 2 | 1.5065E+05/7.8378E+04(-) | 3.7940E+04/2.6191E+04(-) | 8.0899E+03/7.5145E+03(-) | 5.5515E+03/5.2247E+03 |
| 3 | 6.8292E+05/1.3908E+06(-) | 1.9546E+05/8.0087E+05(-) | 6.0980E+04/4.3308E+05(=) | 3.0852E+05/9.8151E+05 |
| 4 | 2.4935E+01/2.4692E+01(+) | 8.7166E+00/1.1532E+01(+) | 2.0449E-06/9.0193E-06(+) | 1.0898E+04/8.1146E+03 |
| 5 | 1.0031E-13/3.6993E-14(-) | 0.0000E+00/0.0000E+00(+) | 1.1369E-13/0.0000E+00(-) | 3.3437E-14/5.2316E-14 |
| 6 | 1.2582E+01/3.6330E+00(-) | 3.0751E+00/2.8582E+00(-) | 1.5534E+00/6.2753E+00(+) | 1.0356E+00/5.1769E+00 |
| 7 | 2.6302E+00/2.1628E+00(=) | 5.0614E+00/4.2506E+00(-) | 3.5178E+00/3.2638E+00(-) | 2.3869E+00/2.2740E+00 |
| 8 | 2.0939E+01/5.6807E-02(=) | 2.0949E+01/5.1278E-02(=) | 2.0786E+01/2.0506E-01(+) | 2.0947E+01/4.9852E-02 |
| 9 | 2.5244E+01/4.9106E+00(-) | 1.5139E+01/5.2516E+00(+) | 2.8462E+01/1.6933E+00(-) | 1.7265E+01/2.7858E+00 |
| 10 | 3.7285E-02/1.9657E-02(=) | 7.4572E-02/5.5460E-02(-) | 2.4286E-02/1.7553E-02(+) | 3.5551E-02/2.1304E-02 |
| 11 | 0.0000E+00/0.0000E+00(=) | 0.0000E+00/0.0000E+00(=) | 3.3437E-15/1.3508E-14(=) | 0.0000E+00/0.0000E+00 |
| 12 | 6.0004E+01/1.0484E+01(-) | 3.6587E+01/1.1787E+01(-) | 1.5091E+01/2.6538E+00(=) | 1.7714E+01/7.2898E+00 |
| 13 | 8.8767E+01/1.5304E+01(-) | 6.6737E+01/1.9722E+01(-) | 3.1289E+01/1.1550E+01(-) | 2.6830E+01/1.0763E+01 |
| 14 | 2.8576E-03/7.2355E-03(+) | 5.7645E+01/2.0876E+01(-) | 1.1430E-02/1.4005E-02(+) | 1.7682E-01/4.7684E-02 |
| 15 | 5.2690E+03/3.9840E+02(-) | 5.9452E+03/2.4500E+02(-) | 3.0519E+03/3.3287E+02(-) | 2.5980E+03/5.4586E+02 |
| 16 | 2.3339E+00/2.4729E-01(-) | 2.3020E+00/2.5910E-01(-) | 8.1608E-01/2.2588E-01(+) | 1.5029E+00/8.5180E-01 |
| 17 | 3.0434E+01/1.6979E-10(=) | 3.5004E+01/1.1533E+00(-) | 3.0434E+01/6.2106E-14(=) | 3.0434E+01/2.4964E-10 |
| 18 | 1.5786E+02/1.4910E+01(-) | 1.5775E+02/9.7803E+00(-) | 6.3599E+01/3.8715E+00(-) | 4.3870E+01/5.9574E+00 |
| 19 | 1.6687E+00/1.2313E-01(-) | 3.9242E+00/3.6905E-01(-) | 1.1367E+00/8.8781E-02(-) | 9.7159E-01/1.6784E-01 |
| 20 | 1.1794E+01/3.2349E-01(-) | 1.1074E+01/3.4347E-01(-) | 1.0798E+01/8.3008E-01(-) | 9.8922E+00/6.3677E-01 |
| 21 | 2.7596E+02/6.8258E+01(+) | 3.1894E+02/8.2409E+01(=) | 3.0231E+02/5.7053E+01(=) | 3.1075E+02/6.6173E+01 |
| 22 | 1.1878E+02/1.6641E+01(-) | 2.8270E+02/1.7820E+02(-) | 9.1992E+01/3.3980E+01(=) | 1.0516E+02/1.3415E+01 |
| 23 | 5.1992E+03/5.4200E+02(-) | 6.0993E+03/4.2019E+02(-) | 3.2695E+03/3.9609E+02(-) | 2.6564E+03/5.7735E+02 |
| 24 | 2.1206E+02/9.3564E+00(-) | 2.1229E+02/4.7826E+00(-) | 2.1097E+02/7.0473E+00(-) | 2.0666E+02/5.2553E+00 |
| 25 | 2.4993E+02/1.0268E+01(+) | 2.3885E+02/2.3635E+01(+) | 2.5487E+02/1.6331E+01(+) | 2.5856E+02/6.6776E+00 |
| 26 | 2.0230E+02/1.6375E+01(-) | 2.0000E+02/1.2099E-03(-) | 2.1258E+02/3.4746E+01(=) | 2.1130E+02/3.4813E+01 |
| 27 | 6.6274E+02/1.9553E+02(-) | 4.2422E+02/4.2415E+01(-) | 4.1820E+02/1.4126E+02(=) | 4.1780E+02/1.4498E+02 |
| 28 | 3.0000E+02/6.4311E-14(=) | 3.0000E+02/0.0000E+00(=) | 3.0000E+02/0.0000E+00(=) | 3.0000E+02/0.0000E+00 |
| -/=/+ | 17/7/4 | 19/5/4 | 12/9/7 | -/-/- |

improvement in 26 out of 28 functions in comparing with SaDE, improvement in 17 out of 28 functions in comparing with SHADE. All of the contrasted algorithms can get the global optima on F1, F11. CIPDE, CoDE, JADE, jDE, SaDE, and CIpBDE algorithms can get the global optima on F5. Only CobiDE and SHADE cannot get the global optimum on F5. For the other benchmark functions, all the contrasted algorithms cannot find the global optima.

As shown in Table 3, our proposed CIpBDE outperforms the contrasted algorithms from the mean and the standard deviation perspective of view. Our proposed CIpBDE algorithm exhibits either better or similar performance improvement in 26 out of 28 functions in comparing with CIPDE, improvement in 24 out of 28 functions in comparing with CobiDE, improvement in 22 out of 28 functions in comparing with CoDE, improvement in 26 out of 28 functions in comparing with JADE, improvement in 24 out of 28 functions in comparing with jDE, improvement in 24 out of 28 functions in comparing with SaDE, improvement in 21 out of the total 28 functions in comparing with SHADE. In a word, our CIpBDE algorithm outperforms these state-of-the-art DE variants.

# 6 Our Proposed CIpBDE Algorithm for the Feature Selection Problem

In this part, we present CIpBDE algorithm for the feature selection problem. In recent years, feature selection has attracted a lot of attention [52-54] since it can enhance prediction accuracy and reduce the computational cost of data mining. Feature selection is an essential step adopted in various tasks, such as data mining, cluster analysis, classification, pattern recognition. Its purpose is to eliminate the useless features without reducing the prediction accuracy, and extract the useful feature subset from the original feature set [55]. However, find useful feature subset is a challenging task because of the huge search space and the complicated interaction among features. There are $2^n$ possible subsets for a given dataset having $n$ features. Searching all possible solutions for a large $n$ is impractical because it is too costly and restrictive. Therefore, feature selection can be considered as an NP-hard problem [56]. Concerning this issue, Some meta-heuristic search algorithms [52-54] were applied to solve this problem. In this study, we apply CIpBDE to solve the feature selection problem and make a comparison with GA, PSO, DE, and ABC algorithms.

## 6.1 Encoding of the Individual

In this paper, we adopt the numerical encode of CIpBDE to solve the feature selection problem. Each individual indicates the total number of features of a given dataset and each element denotes the probability of a corresponding feature to be selected. For example, for a given data set with D features, the individual with D dimensions is given as fellows.

$$X_i = (\mathrm{x}_{i,1}, \mathrm{x}_{i,2}, ..., \mathrm{x}_{i,\mathrm{D}}),\ i = 1, 2, ..., \mathrm{NP} \qquad \textbf{(22)}$$

where $NP$ is the population size, $x_{i,j} \in [0,1]$ is the probability of choosing the $j$th feature in the feature subset. Each individual is decoded to binary string with a user-defined threshold $\eta$. When the value of the $j$th element of individual is greater than $\eta$, the corresponding $j$th feature is selected to the feature subset and the corresponding decoded element is 1. Otherwise, the corresponding feature is discarded and the corresponding decoded element is 0. An example of feature subset solution for a given dataset that has 14 features is given in Figure 3. In this paper, the threshold value of $\eta$ is set to 0.5.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |

**Figure 3.** An example of feature subset solution

## 6.2 Fitness Evaluation

The goal of feature selection is to determine a feature subset with a strong classification capability. In this paper, we apply the one nearest neighbor (1-NN) algorithm as a classifier to evaluate feature subsets that are represented by the individuals. 1-NN is a special case of k-nearest neighbor (KNN) [57] when k is set to 1. The reason we chose KNN algorithm is that KNN is very popular and easy to implement. On the other hand, a ten-fold cross-validation is adopted to improve the reliability of performance evaluation. In k-fold cross-validation, the given data set is randomly divided into k equal-sized folds. One of them is selected as the testing set, and the rest k-1 folds constitute the training set. This process is repeated k times so that each fold is used once as the testing set.

The performance of each fold is measured with the accuracy evaluation measure, i.e., the percentage of samples that are correctly classified, as given in Equation (23).

$$Accuracy = \frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ all\ the\ samples} \qquad \textbf{(23)}$$

The overall performance of the classification is the average of the k results obtained by all folds, as given in Equation (24) which is also the fitness function of the individual. The higher the average accuracy, the better is the performance of feature subset, i.e., the corresponding individual has better fitness value.

$$fitness = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \qquad \textbf{(24)}$$

The whole flowchart of CIpBDE for feature selection problem is illustrated in Figure 4.

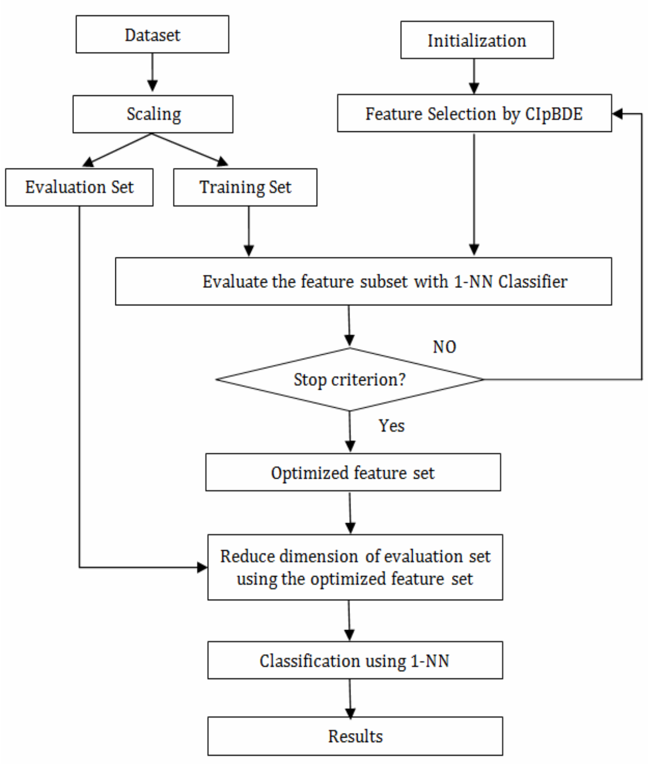


**Figure 4.** The Flowchart of the CIpBDE for feature selection

### 6.3 Experiments Analysis for Feature Selection

In this subsection, we evaluate the effectiveness of the CIpBDE for feature selection problem on several well-known real-world data sets from different knowledge fields, including Glass, Heart, Ionosphere, Iris, Parkinson, Segmentation, Sonar, Vowel, and Wine. These data sets are cited from the UCI Machine Learning Repository [58]. We compare the experimental results of CIpBDE with GA, PSO, DE, and ABC algorithms. The population size of all comparison algorithms is set to 20. The maximum iteration number is set to 100. The other parameters of CIpBDE are the same as section 5. The parameter settings for the other comparison algorithms are listed in Table 4. The final performance of all the experimental results are the average of 10 runs independently experiments. The comparison results are collected in Table 5.

**Table 4.** Parameter settings for experiments

| Algorithm | Parameters settings |
|---|---|
| GA | *crossover rate* = 0.8, *mutation rate* = 0.05 |
| PSO | $c_1 = c_2 = 2, w_{max} = 0.9, w_{min} = 0.4$ |
| DE | $F = 0.7, Cr = 0.1$ |
| ABC | FoodNumber=10, limit=10 |

**Table 5.** Results for five comparison algorithms on 9 UCI datasets

| Data Set | GA | | PSO | | DE | | ABC | | CIpBDE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Average feature | Average accuracy | Average feature | Average accuracy | Average feature | Average accuracy | Average feature | Average accuracy | Average feature | Average accuracy |
| Glass | 5.7 | 80.61% | 5.1 | 81.08% | 5.8 | 81.42% | 5.6 | 79.91% | 6 | 81.54% |
| Heart | 7.8 | 83.26% | 8.4 | 83.85% | 9 | 84.19% | 7.7 | 82.96% | 8.3 | 84.70% |
| Ionosphere | 13.3 | 93.33% | 13.5 | 93.91% | 11.4 | 94.42% | 14.8 | 92.40% | 11.8 | 94.71% |
| Iris | 2.1 | 97.80% | 2.3 | 97.73% | 2 | 98.00% | 2 | 97.60% | 2.1 | 98.00% |
| Parkinson | 11.8 | 99.13% | 11.9 | 99.33% | 12.2 | 99.49% | 12.8 | 98.56% | 12 | 99.90% |
| Segmentation | 11 | 97.89% | 11.6 | 97.94% | 11.1 | 98.00% | 11.1 | 97.59% | 11.1 | 98.04% |
| Sonar | 30.7 | 92.74% | 30.7 | 94.29% | 28.7 | 93.56% | 29.5 | 91.30% | 30 | 94.71% |
| Vowel | 9.1 | 99.50% | 8.8 | 99.37% | 9.3 | 99.39% | 8.9 | 99.09% | 9.1 | 99.52% |
| Wine | 8.4 | 99.39% | 8.8 | 99.50% | 8.1 | 99.67% | 7.6 | 99.22% | 7.9 | 99.67% |

From the Table 5, we can observe that the CIpBDE algorithm obtains the best classification accuracy among the comparison algorithms. The classification accuracy achieved by the CIpBDE algorithm is 81.54%, 84.70%, 94.71%, 98.00%, 99.90%, 98.04%, 94.71%, 99.52%, and 99.67% on data sets Glass, Heart, Ionosphere, Iris, Parkinson, Segmentation, Sonar, Vowel, and Wine, respectively. For the data sets Iris and Wine, the CIpBDE and DE algorithms have the same performance. For the average size of the selected feature subset, the CIpBDE algorithm has no minimum size. It should notice that the goal of feature selection is to eliminate the useless features without reducing the classification accuracy, which means that the smallest or largest feature subset size is not equivalent to the best or worst classification accuracy.

## 7 Conclusions

In this paper, a novel DE variant was proposed, namely CIpBDE, based on hybridizing the top collective information and p-best information for global optimization problems. In the proposed CIpBDE, first, we proposed a combined mutation strategy CIpBM by taking advantage of the mutation strategies "target-to-ci_pbest/1" and "target-to-pbest/1" to avoid trapping local optima. Second, we presented a modified crossover operation CIpBX to against the stagnation of DE. The CIpBX uses a collective vector or top p-best individual based on probability to execute crossover operation when stagnation occurs. Finally, we modified the parameter adaptation strategy to tune the parameters and values in each generation.

The performance of CIpBDE is assessed on the CEC2013 benchmark test suite with 28 functions. The experimental results indicate that CIpBDE gives better performance than seven powerful and popular DE variants. To further evaluate the effectiveness of CIpBDE on a real-world problem, we applied CIpBDE to the feature selection problem. The experimental results on several standard data sets demonstrate that the proposed CIpBDE algorithm outperforms the four comparing algorithms in terms of classification accuracy. For the future work, we will test the proposed CIpBDE algorithm on other real-world optimization problems [59-63].

## References


[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[2] J. Kennedy, R. Eberhart, Particle swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942-1948.

[3] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, H.-J. Kim, An Improved Routing Schema with Special Clustering Using PSO Algorithm for Heterogeneous Wireless Sensor Network, *Sensors*, Vol. 19, No. 3. pp. 1-17, February, 2019.

[4] D. Karaboga, B. Basturk, A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459-471, November, 2007.

[5] L. Cui, K. Zhang, G. Li, X. Fu, Z. Wen, N. Lu, J. Lu, Modified Gbest-guided Artificial Bee Colony Algorithm with New Probability Model, *Soft Computing*, Vol. 22, No. 7, pp. 2217-2243, April, 2018.

[6] S. C. Chu, P. W. Tsai, J. S. Pan, Cat Swarm Optimization, in: Q. Yang, G. Webb (Eds.), *PRICAI 2006: Trends in Artificial Intelligence, Lecture Notes in Computer Science*, Vol. 4099, Springer, Berlin, Heidelberg, 2006, pp. 854-858.

[7] X.-F. Ji, J.-S. Pan, S.-C. Chu, P. Hu, Q.-W. Chai, P. Zhang, Adaptive Cat Swarm Optimization Algorithm and Its Applications in Vehicle Routing Problems, *Mathematical Problems in Engineering*, Vol. 2020, pp. 1-14, April, 2020.

[8] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341-359, December, 1997.

[9] Z. Meng, J. S. Pan, K. K. Tseng, PaDE: An Enhanced Differential Evolution Algorithm with Novel Control Parameter Adaptation Schemes for Numerical Optimization, *Knowledge-Based Systems*, Vol. 168, pp. 80-99, March, 2019.

[10] X. Sui, S. C. Chu, J. S. Pan, H. Luo, Parallel Compact Differential Evolution for Optimization Applied to Image Segmentation, *Applied Sciences*, Vol. 10, No. 6, pp. 1-22, March, 2020.

[11] J. S. Pan, N. Liu, S. C. Chu, A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning, *IEEE Access*, Vol. 8, pp. 17691-17712, January, 2020.

[12] T. K. Dao, S. C. Chu, J. S. Pan, T.-T. Nguyen, T.-G. Ngo, T.-D. Nguyen, H.-T. Tran, An improved bat algorithm Based on Hybrid with Ant Lion Optimizer, in: J. S. Pan, J. W. Lin, Y. Liang, S. C. Chu (Eds.), *Advances in Intelligent Systems and Computing*, Vol. 1107 AISC, Springer, Singapore, 2020, pp. 50-60.

[13] S. Das, P. N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art, *IEEE Transactions on Evolutionary Computation*, Vol. 15, No.1, pp. 4-31, February, 2011.

[14] S. Das, S. S. Mullick, P. N. Suganthan, Recent Advances in Differential Evolution – An Updated Survey, *Swarm & Evolutionary Computation*, Vol. 27, pp. 1-30, April, 2016.

[15] X. Xue, J. Chen, D. Chen, Matching Biomedical Ontologies through Compact Hybrid Evolutionary Algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 10, No. 1, pp. 110-117, January, 2019.

[16] K. Price, R. M. Storn, J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*, Springer-Verlag Berlin Heidelberg, 2005.

[17] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, May, 1983.

[18] Z. Meng, J. S. Pan, H. Xu, QUasi-Affine TRansformation Evolutionary (QUATRE) Algorithm: A Cooperative Swarm Based Algorithm for Global Optimization, *Knowledge-Based Systems*, Vol. 109, pp. 104-121, October, 2016.

[19] Z. Meng, J. Pan, HARD-DE: Hierarchical ARchive Based Mutation Strategy With Depth Information of Evolution for the Enhancement of Differential Evolution on Numerical Optimization, *IEEE Access*, Vol. 7, pp. 12832-12854, January, 2019.

[20] Z. Meng, C. Yang, X. Li, Y. Chen, Di-DE: Depth Information-Based Differential Evolution with Adaptive Parameter Control for Numerical Optimization, *IEEE Access*, Vol. 8, pp. 40809-40827, February, 2020.

[21] J. Zhang, A. C. Sanderson, JADE: Adaptive Differential Evolution With Optional External Archive, *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 5, pp. 945-958, October, 2009.

[22] L. M. Zheng, S. X. Zhang, K. S. Tang, S. Y. Zheng, Differential Evolution Powered by Collective Information, *Information Sciences*, Vol. 399, pp. 13-29, August, 2017.

[23] L. Cui, G. Li, Z. Zhu, Q. Lin, K. Wong, J. Chen, N. Lu, J. Lu, Adaptive Multiple-elites-guided Composite Differential Evolution Algorithm with a Shift Mechanism, *Information Sciences*, Vol. 422, pp. 122-143, January, 2018.

[24] S. M. Islam, S. Das, S. Ghosh, S. Roy, P. N. Suganthan, An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization, *IEEE Transactions on Systems, Man, and*

*Cybernetics. Part B, Cybernetics*, Vol. 42, No. 2, pp. 482-500, April, 2012.

[25] M. Pant, M. Ali, V. P. Singh, Differential Evolution with Parent Centric Crossover, *EMS '08. Second UKSIM European Symposium on Computer Modeling and Simulation*, Liverpool, UK, 2008, pp. 141-146.

[26] Z. Meng, J. S. Pan, QUasi-affine TRansformation Evolutionary (QUATRE) Algorithm: A Parameter-reduced Differential Evolution Algorithm for Optimization Problems, *Proceedings IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 2016, pp. 4082-4089.

[27] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems, *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 6, pp. 646-657, December, 2006.

[28] N. H. Awad, M. Z. Ali, P. N. Suganthan, Ensemble of Parameters in a Sinusoidal Differential Evolution with Niching-based Population Reduction, *Swarm and Evolutionary Computation*, Vol. 39, pp. 141-156, April, 2018.

[29] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp. 398-417, April, 2009.

[30] R. Storn, Differential Evolution Research—Trends and open Questions, in: U. K. Chakraborty (Ed.), *Advances in Differential Evolution*, Springer, Berlin, Heidelberg, 2008, pp. 1-31.

[31] Z. Meng, J. S. Pan, L. Kong, *Parameters with Adaptive Learning Mechanism (PALM) for the Enhancement of Differential Evolution*, Knowledge-Based Systems, Vol. 141, pp. 92-112, February, 2018.

[32] J. Lampinen, I. Zelinka, On Stagnation Of The Differential Evolution Algorithm, Proceedings of *MENDEL 2000, 6th International Mendel Conference on Soft Computing*, Brno, Czech Republic, 2000, pp. 76-83.

[33] M. Yang, C. Li, Z. Cai, J. Guan, Differential Evolution with Auto-enhanced Population Diversity, *IEEE Transactions on Cybernetics*, Vol. 45, No. 2, pp. 302-315, February, 2015.

[34] L. Cui, Q. Huang, G. Li, S. Yang, Z. Ming, Z. Wen, N. Lu, J. Lu, Differential Evolution Algorithm with Tracking Mechanism and Backtracking Mechanism, *IEEE Access*, Vol. 6, pp. 44252-44267, August, 2018.

[35] H. Y. Fan, J. Lampinen, A Trigonometric Mutation Operation to Differential Evolution, *Journal of Global Optimization*, Vol. 27, No. 1, pp. 105-129, September, 2003.

[36] A. W. Mohamed, An Improved Differential Evolution Algorithm with Triangular Mutation for Global Numerical Optimization, *Computers & Industrial Engineering*, Vol. 85, pp. 359-375, July, 2015.

[37] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential Evolution Using a Neighborhood-based Mutation Operator, *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 3, pp. 526-553, June, 2009.

[38] W. Yu, M. Shen, W. Chen, Z. Zhan, Y. Gong, Y. Lin, O. Liu, J. Zhang, Differential Evolution with Two-Level Parameter Adaptation, *IEEE Transactions on Cybernetics*, Vol. 44, No. 7, pp. 1080-1099, July, 2014.

[39] Y. Wang, Z. Cai, Q. Zhang, Enhancing the Search Ability of Differential Evolution through Orthogonal Crossover, *Information Sciences*, Vol. 185, No. 1, pp. 153-177, February, 2012.

[40] Y. Wang, H. X. Li, T. Huang, L. Li, Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting, *Applied Soft Computing*, Vol. 18, pp. 232-247, May, 2014.

[41] S. M. Guo, C. C. Yang, Enhancing Differential Evolution Utilizing Eigenvector-based Crossover Operator, *IEEE Transactions on Evolutionary Computation*, Vol. 19, No. 1, pp. 31-49, February, 2015.

[42] R. Tanabe, A. Fukunaga, Success-history Based Parameter Adaptation for Differential Evolution, *2013 IEEE Congress on Evolutionary Computation,* Cancun, Mexico, 2013, pp. 71-78.

[43] R. Tanabe, A. S. Fukunaga, Improving the Search Performance of SHADE Using Linear Population Size Reduction, *2014 IEEE Congress on Evolutionary Computation*, Beijing, China, 2014, pp. 1658-1665.

[44] A. Draa, S. Bouzoubia, I. Boukhalfa, A Sinusoidal Differential Evolution Algorithm for Numerical Optimisation, *Applied Soft Computing*, Vol. 27, pp. 99-126, February, 2015.

[45] Y. Wang, Z. Cai, Q. Zhang, Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters, *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 55-66, February, 2011.

[46] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies, *Applied Soft Computing*, Vol. 11, No. 2, pp. 1679-1696, March, 2011.

[47] R. Mallipeddi, Harmony Search Based Parameter Ensemble Adaptation for Differential Evolution, *Journal of Applied Mathematics*, Vol. 2013, pp. 1-12, July, 2013.

[48] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, H. Chen, Differential Evolution with Multi-population Based Ensemble of Mutation Strategies, *Information Sciences*, Vol. 329, pp. 329-345, February, 2016.

[49] S. M. Guo, C. C. Yang, P. H. Hsu, J. S. H. Tsai, Improving Differential Evolution with a Successful-parent-selecting Framework, *IEEE Transactions on Evolutionary Computation*, Vol. 19, No. 5, pp. 717-730, October, 2015.

[50] Y. Zhou, J. Wang, Y. Zhou, Z. Qiu, Z. Bi, Y. Cai, Differential Evolution with Guiding Archive for Global Numerical Optimization, *Applied Soft Computing*, Vol. 43, pp. 424-440, June, 2016.

[51] J. J. Liang, B. Y. Qu, P. N. Suganthan, A. G. Hernández-Díaz, *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-parameter Optimization*, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical report 201212, January, 2013.

[52] E. Hancer, B. Xue, M. Zhang, Differential Evolution for Filter Feature Selection Based on Information Theory and

Feature Ranking, *Knowledge-Based Systems*, Vol. 140, pp. 103-119, January, 2018.

[53] M. Schiezaro, H. Pedrini, Data Feature Selection Based on Artificial Bee Colony Algorithm, *EURASIP Journal on Image and Video Processing*, Vol. 2013, No. 1, Article number: 47, August, 2013.

[54] Y. Shi, C. M. Pun, H. Hu, H. Gao, An Improved Artificial Bee Colony and Its Application, *Knowledge-Based Systems*, Vol. 107, pp. 14-31, September, 2016.

[55] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature Selection Algorithm Based on Bare Bones Particle Swarm Optimization, *Neurocomputing*, Vol. 148, pp. 150-157, January, 2015.

[56] A. Unler, A. Murat, R. B. Chinnam, Mr2PSO: A Maximum Relevance Minimum Redundancy Feature Selection Method Based on Swarm Intelligence for Support Vector Machine Classification, *Information Sciences*, Vol. 181, No. 20, pp. 4625-4641, October, 2011.

[57] T. M. Cover, P. E. Hart, Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21-27, January, 1967.

[58] P. M. Murphy, D. W. Aha, UCI Repository of Machine Learning Databases, *University of California*, http://archive.ics.uci.edu/ml/.

[59] S. Wang, F. Sun, J.-S. Pan, H. Zhang, A Novel Method for CCSDS-TC Recognition, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 313-322, March, 2020.

[60] S.-C. Chu, X. Xue, J.-S. Pan, X. Wu, Optimizing Ontology Alignment in Vector Space, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 15-22, January, 2020.

[61] S. Zhang, X. Su, X. Jiang, M. Chen, T.-Y. Wu, A Traffic Prediction Method of Bicycle-sharing based on Long and Short term Memory Network, *Journal of Network Intelligence*, Vol. 4, No. 2, pp. 17-29, May, 2019.

[62] T. Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, C. M. Chen, An Authenticated Key Exchange Protocol for Multi-Server Architecture in 5G Networks, *IEEE Access*, Vol. 8, pp. 28096-28108, January, 2020.

[63] C. M. Chen, W. Fang, S. Liu, T. Y. Wu, J. S. Pan, K. H. Wang, Improvement on a Chaotic Map-based Mutual Anonymous Authentication protocol, *Journal of Information Science and Engineering*, Vol. 34, No. 2, pp. 371-390, March, 2018.

## Biographies

**Nengxian Liu** is a Ph.D. student in Fuzhou University. His current research interests include evolutionary algorithms and their applications in real-world problems.

**Jeng-Shyang Pan** received the Ph.D. degree in Electrical Engineering from the University of Edinburgh, U.K. in 1996. Currently, he is a Professor in Shandong University of Science and Technology and Ph.D. advisor in the Fuzhou University. His current research interests include soft computing, information security and signal processing.

**Jinfeng Lai** is a professor in the School of Information and Communication Engineering, University of Electronic Science and Technology of China. His research directions include embedded systems, sensor-based healthcare, and multimedia communications. He is currently an editorial board member of several international SCI journals. He is an IEEE Senior Member.

**Shu-Chuan Chu** received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. Currently, she is the Research Fellow with PhD advisor in the Shandong University of Science and Technology. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.

**Trong-The Nguyen** received his Ph.D. degree in 2016 from the National Kaohsiung University of Applied Sciences, Taiwan. He is currently a lecturer in the Fujian University of Technology, China and Haiphong University of Manager and Technology, Vietnam. His current research interests include computational intelligence, signal processing, and sensor networks.