

Article

# Application of Differential Evolution Algorithm Based on Mixed Penalty Function Screening Criterion in Imbalanced Data Integration Classification

Yuelin Gao <sup>1,\*</sup> , Kaiguang Wang <sup>1,\*</sup> , Chenyang Gao <sup>2</sup>, Yulong Shen <sup>2</sup> and Teng Li <sup>2</sup>

<sup>1</sup> Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Yinchuan 750021, China

<sup>2</sup> School of Cyber Engineering, Xidian University, Xi'an 710071, China; ghygao2008@gmail.com (C.G.); ylshen@mail.xidian.edu.cn (Y.S.); tengli@xidian.edu.cn (T.L.)

\* Correspondence: gaoyuelin@263.net (Y.G.); wkg13759842420@foxmail.com (K.W.); Tel.: +86-139-9510-0900 (Y.G.); +86-183-2997-0138 (K.W.)

Received: 20 November 2019; Accepted: 10 December 2019; Published: 13 December 2019



**Abstract:** There are some processing problems of imbalanced data such as imbalanced data sets being difficult to integrate efficiently. This paper proposes and constructs a mixed penalty function data integration screening criterion, and proposes Differential Evolution Integration Algorithm Based on Mixed Penalty Function Screening Criteria (DE-MPFSC algorithm). In addition, the theoretical validity and the convergence of the DE-MPFSC algorithm are analyzed and proven by establishing the Markov sequence and Markov evolution process model of the DE-MPFSC algorithm. In this paper, the entanglement degree and entanglement degree error are introduced to analyze the DE-MPFSC algorithm. Finally, the effectiveness and stability of the DE-MPFSC algorithm are verified by UCI machine learning datasets. The test results show that the DE-MPFSC algorithm can effectively improve the effectiveness and application of imbalanced data classification and integration, improve the internal classification of imbalanced data and improve the efficiency of data integration.

**Keywords:** imbalanced data; screening criteria; DE-MPFSC algorithm; Markov process; entanglement degree; data integration

**MSC:** 90C59; 54A05

## 1. Introduction

In data classification, the most common methods used are clustering methods based on statistical theory [1] such as naive Bayes [2] and artificial neural networks (ANNs) [3]. These methods can use large-scale imbalanced data as clustering targets [4]. However, with the development of large data networks, electronic data integration, imbalanced data analysis, large text databases, quantum data decoding and encoding analysis, and other data types in recent years, the efficient processing of data has become the foremost subject of current data classification and processing. The imbalanced data classification system studied in this paper belongs to this subject. When the imbalanced data is classified and processed, integration and classification of the data has gradually become the mainstream method for processing imbalanced data but this method usually results in a higher computational cost [5]. In view of the cause, we have identified a meaningful conclusion—the efficient clustering of imbalanced data processing tends to increase the processing cost.

For the processing of imbalanced data, there are some mature algorithms such as sampling methods [6,7], cost-sensitive algorithms [8,9] and one-class classification [8,9]. However, there are some problems in that the processing of imbalanced data is mainly confined to the level of computer

algorithm analysis and big data analysis, and lacks the function of efficient integration, which can result in privacy or data breaches, resulting in significant losses to data owners. Distributed and compatible construction of evolutionary algorithm structures is an innovative idea, which was proposed by Santucci [10]. Centroid classification algorithm will provide new ideas for us to explore and solve imbalanced data classification problems. Mikalef proposed a new idea about data processing. Application level and resource level of big data can improve investment efficiency and business performance, which idea was proposed by Patrick Mikalef and it is applied preferentially to business [11]. In addition to the above algorithms, there are other scholars who have performed research on imbalanced data processing. To seek parameter optimization to reduce the cost before training the model, Thai-Nghe proposed a fault-tolerant recognition algorithm that sets the artificial cost rate of data processing to ultrahigh parameters [12]. In the integration of imbalanced data, most experts focus on how to improve the efficient integration and the integration of imbalanced data, such as the boosting data analysis method [13] mentioned by Freund and the bagging data integration analysis algorithm [14] proposed by Breiman. Sun proposed a modular data analysis integration strategy that classifies and integrates multiple imbalanced data and analyzed data using modular strategies [15]. Chawla proposed a data integration strategy combining the (Synthetic Minority Oversampling Technique) SMOTE algorithm and boosting algorithm used in the sampling process [16]. Lachiche [17] proposed a new effective recursive algorithm combining 1BC and 2BC Bayesian structural systems, improving the difference between computer systems and artificial data in low-dimensional levels.

For generalized imbalanced data processing, the introduction of the above algorithms is limited to the specific algorithm program of data integration and the results of data classification. However, for the number of feature samples and the feature categories about the imbalanced data, the above algorithms may cause higher data classification costs and classification errors. Overall, in the previous research methods, most of them are integrated strategies under a single integrated classifier structure model such as support vector machine, decision tree analysis and the naive Bayesian method [18]. These methods determine the integration effect based on data performance as a classification criterion, which is not conducive to the continuous integration and classification of imbalanced datasets. To solve the problem of high-efficiency classification of imbalanced data, this paper establishes a differential evolution integration algorithm with global convergence ability [19,20] based on the screening criterion of the mixed penalty function (differential evolution integration algorithm based on mixed penalty function screening criteria—the DE-MPFSC algorithm), which provides a useful method for solving the high-efficiency integration and classification of imbalanced data.

### **A: Source of The Idea**

The idea of the DE-MPFSC algorithm is derived from the following three excellent overview papers. The purpose of the first paper [21] is to summarize and organize the comprehensive foundation and the recent proposals on parameter adaptation about differential evolution (DE), which inspired us that the optimal parameters analysis should be carried out before the algorithm design. The second paper [22], presents the basic structure, parameter analysis and the latest applications of evolutionary algorithms. It describes the related principles of various evolutionary algorithms in detail. Comparing the performance of DE algorithm is an effective way to further improve the understanding of evolutionary algorithm. The third paper [23] introduces the various structures and main variants of the DE algorithm. The applications of the DE algorithm in the fields of commerce, agriculture and economics was emphasized in the literature. The theoretical design in this paper [23] is consistent with the ideas about the topological structure of the divided regions [20]. Because the population is so sensitive to the spatial structure of the search area during the evolution process, we carefully considered this idea and proposed a screening criterion based on region segmentation. However, when we reviewed their contributions, we had one common finding: for the variant of DE, when we test the algorithm, we implement it based on a procedural evaluation system of test functions and

dimensions. However, the evaluation system ignores the important influence of convergence speed and convergence precision on individual evolution. Based on this finding, we systematically discuss the influence of convergence precision and convergence speed on the evolution process based on the structural indicators of the evolutionary algorithm. Finally, we propose an important idea: judge whether the evolutionary algorithm is better than before based on the degree of entanglement between convergence precision and convergence speed, then, further verify the scientificity of the idea proposed by us through numerical experiments.

In terms of improvements in the DE algorithm, in a multimodal interactive environment, balancing the parameters  $F$  and  $CR$  of the evolutionary algorithm is an important idea. In multi-objective parameter design and adaptive parameter analysis, the idea of self-adaptation was proposed by Janez Brest, namely (Self-Adapting Parameter Setting in Differential Evolution)JDE [21,24] and applied in the field of computer and machine learning, which can be also applied to large-scale problems [25]. The dynamic parameter design idea of DE-MPFSC algorithm is based on this research and numerical experiments are also analyzed in detail. The Opposition Based Differential Evolution (OBDE), proposed by Rahnamayan et al. (2008) [22,26], mainly emphasizes the impact of the evolution speed on the DE algorithm. The OBDE algorithm can enhance the adaptability of the algorithm by improving the learning efficiency, which idea has a priority role in solving multi-objective problems. Our inspiration for the idea of decision space segmentation in this paper comes from it, which is innovative for improving the decision structure of evolutionary algorithms. The (Differential Evolution with Global and Local neighborhoods) DEGL algorithm [23,27] has made some improvements in the individual optimization space, emphasizing the influence of topological structure and spatial neighborhood on the evolutionary algorithm and highlighting the role of the idea of spatial segmentation, which can guide us to think deeply about the search area of the evolutionary algorithm [28,29]. The new model will be widely used in technological improvement. In Reference [20], we independently proposed the spatial topology concept of the DE algorithm and gave relevant proofs. We independently proposed the spatial topology concept of the DE evolution algorithm and gave relevant proofs. Topology concept is a priority direction in expanding search area. Qin and Suganthan proposed a new version of adaptive differential evolution algorithm [22,30]. Through the adaptation of mutation factor  $F$  and cross probability  $CR$ , the (Self-Adaptive Differential Evolution) SADE algorithm can adjust the individual search precision and search speed from a micro perspective, which will fundamentally change the search pattern of individuals in the spatial neighborhood and will promote the healthy development of the DE algorithm and provide useful help for the dynamic adjustment strategy of the DE-MPFSC algorithm. In addition, the Modified DE algorithm (MDE) was proposed by Zou et al. [21,31]. The main idea of MDE is to mix the adaptive mechanism of mean and Gaussian distribution to improve the ability of parameters  $F$  and  $CR$  to update themselves. The Fitness Adaptation Differential Evolution (FADE) was proposed by Ghosh et al. [21,32], the strategy selection mechanism of the main idea is that individual selection behaviors can be randomly generated during the evolution process. When individual populations evolve in spatial neighborhoods, the tentative strategy of selecting individuals is also an effective way to expand the diversity of the population. The spatial structure of individual evolution is not completely random. The evolutionary pattern of individuals will be affected by mutations, resulting in structural deviations of individuals, which is not conducive to large-scale population evolution [33], which was proposed by Caraffini. The structured properties of evolutionary algorithms is one of the important issues we will explore in the future. Especially in terms of the convergence and stability of evolutionary algorithms, this effect is significant. To this end, we should explore new ideas to reduce structural deviations. We roughly reviewed the variants of the DE algorithm and related research work from recent years. Based on the above viewpoints, we have developed an improved model of the DE algorithm and proposed the DE-MPFSC algorithm, which opens up new idea for further research on the DE algorithm and related applications.

## B: Technical Route

- a. According to the idea of region segmentation, propose the region screening criteria for individuals in the evolution process;
- b. Construct the operation mechanism of the DE-MPFSC algorithm, including the introduction of the dynamic mutation factor  $F$  and the dynamic crossover probability  $CR$ ;
- c. According to the structural evaluation system, the idea about entanglement degree of the convergence speed and convergence precision of the algorithm is proposed;
- d. Compare the four classic DE algorithms JDE, OBDE, DEGL and SADE to further verify the advantages of the DE-MPFSC algorithm.

The first advantage of the DE-MPFSC algorithm is that it can classify and extract the original imbalanced data, generate an imbalanced data point searching area and balanced data point searching area, further redistribute the data searching unit. The second advantage is that it can clarify the progressive boundaries of imbalanced points and balanced points of the classified conditions. When the mixed penalty function is at a progressive boundary, DE algorithms only search inside or outside all data points and cannot cross data boundaries, which is not conducive to decreasing the time for data integration and classification. However, the DE-MPFSC algorithm can optimize the data structure by using the mixed penalty function. The third advantage can improve the accuracy and global optimization of imbalanced data classification and purification. The fourth advantage can analyze many different types of data structures. Because of the DE-MPFSC algorithm having self-adaptability, it can use the global convergence of heuristic algorithms to greatly shorten the search time of the searching area. The DE-MPFSC algorithm expands the searching space, improving the integration efficiency of imbalanced data, which results in wide adaptability. The full-text structure is arranged as follows:

Part one: The first and second chapters introduce the current situation of imbalanced data integration and classification, the advantage analysis of the DE-MPFSC algorithm and the form and normalization of the internal and external penalty functions.

Part two: The third chapter is mainly the processing of the constraint conditions for the mixed penalty function, the formal construction of the DE-MPFSC algorithm and property analysis of the mixed penalty function screening criteria.

Part three: The fourth chapter is mainly the theoretical analysis of the DE-MPFSC algorithm. The validity and convergence of the DE-MPFSC algorithm are analyzed from the mathematical point of view.

Part four: In the fifth chapter, we creatively introduce the entanglement degree and entanglement degree error to compare the performance of the DE-MPFSC by numerical experiments.

Part five: The sixth chapter mainly establishes verification indicators of the Classification Accuracy (CA), Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) datasets (due to the overall structure of this paper, we list the full name of the datasets in this section. The relevant definition of the datasets is in the data test chapter and the reference is attached). We analyze and verify the theoretical nature of the DE-MPFSC algorithm through the imbalanced data set (Unified Communications Irvine Machine Learning Repository, namely UCI) UCI machine learning data set;

Part six: Gives relevant conclusions.

## 2. Prerequisite Knowledge

The general form of the constrained optimization problem will be expressed as follows:

$$\min f(x); s.t. \begin{cases} g_i \leq 0, i = 1, 2, \dots, j \\ h_i = 0, i = j + 1, j + 2, \dots, m, \end{cases} \quad (1)$$

where  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  are the decision variables of the objective function  $f(x)$ ,  $g_i$  is inequality constraint describing the variable, which role of the inequality constraint is to form the search area in

the feasible domain.  $h_i$  is equality constraint that forms a boundary value condition in the feasible domain, which role is to control the boundary of the search area. We call the feasible solution containing only equality constraints the positive constraint solution where its constraints are active constraints; otherwise, non-active constraints.

### 2.1. Basic Steps of the DE Algorithm

The differential evolutionary (DE) algorithm [19,20], proposed by Storn and Price in 1995 to solve Chebyshev inequalities, is an efficient global optimization algorithm, which adopts floating-point vector coding to search in continuous space [20]. There are higher stability, lower volatility and better convergence than others. The specific form of the DE algorithm is as follows.

#### 2.1.1. Population Initialization

Assume that the population individuals of the DE algorithm is  $\mathbf{X}(t) = (X_1, (X_2, \dots, (X_n))^\top$  [20,34]. Then, population individuals are following:

$$X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t), i = 1, 2, \dots, NP, \tag{2}$$

where  $t$  is the number of iterations and  $NP$  is the number of population [20].

Initialization settings: suppose the dimension of the optimization problem is  $D$ . The maximum number of iterations is  $T$ , then the initialization operation of the DE algorithm is expressed as follows [20]:

$$X_i^0 = (x_{i1}^0, x_{i2}^0, \dots, x_{iD}^0) \tag{3}$$

$$x_{ij}^0 = a_{ij} + rand(0, 1) \cdot (b_{ij} - a_{ij}), i = 1, 2, \dots, NP; j = 1, 2, \dots, D, \tag{4}$$

where,  $a_{ij}, b_{ij} \in R$ .

#### 2.1.2. Individual Mutation

When the individual evolves with the DE algorithm, the mutation sites of the individual originate from the two individuals of the parent  $(x_{i_1}^t, x_{i_2}^t)$  [20] in the  $t$ th generation parental individuals, where  $i_1, i_2 \in NP$ . Then, the differential vector is defined as  $D_{i_{1,2}} = (x_{i_1}^t - x_{i_2}^t)$ . For any vector individual  $X_i^t$ , the mutation operation [35,36] is defined as

$$V_i^{t+1} = X_{i_3}^t + F \cdot (X_{i_1}^t - X_{i_2}^t), \tag{5}$$

where  $NP \geq 4$ ,  $F$  is the mutation factor and  $i_1, i_2, i_3 \in \{1, 2, \dots, NP\}$  and  $i_1, i_2, i_3$  are not all the same during population evolution [20].

#### 2.1.3. Individual Crossover

Test individuals  $U_i^{t+1}$  are generated from mutated individual  $V_i^{t+1}$  and individual  $X_i^t$  before iteration. At the same time, we introduce random function  $rand(0, 1)$  and the crossover probability  $CR$  to improve the diversity and stability of individual evolution and ensure that one mutation site of the test individual  $U_i^{t+1}$  is provided by at least the individual from the last iteration  $(x_i^t)$  or  $V_i^{t+1}$ . The individual crossover [20,36] is following:

$$(u_{ij}^{t+1}) = \begin{cases} (v_{ij}^{t+1}), & \text{if } rand_j(0, 1) \leq CR, \\ (x_{ij}^t), & \text{otherwise.} \end{cases} \quad i = 1, 2, \dots, NP; j = 1, 2, \dots, D, \tag{6}$$

where  $CR = 0$  and  $CR = 1$  are two extreme cases of individual crossover [20]. The former is conducive to maintaining the global optimization ability of the population and the latter is conducive to increasing

the diversity of the population.  $CR \in (0,1)$  is conducive to expanding the global search range of population individuals, increasing population diversity and improving the search precision of the population individuals [20].

### 2.1.4. Individual Selection

The selection operation of the DE algorithm in the evolutionary group is based on the greedy search mechanism. This search mechanism is based on “the individual fitness function value after iteration being smaller than the individual fitness function value” before iteration as a standard [20]. The DE algorithm chooses between test individual  $X_i^t$  and mutation individual  $U_i^{t+1}$  and mutation individual with small fitness function values will be evolved to the next generation [35,37]. The selection effect of the selection operator [20,35,37] in the population is described by the following equation:

$$X_i^{t+1} = \begin{cases} U_i^{t+1}, & \text{if } f(U_i^{t+1}) \leq f(X_i^t), \\ X_i^t, & \text{otherwise.} \end{cases} \quad i = 1, 2, \dots, NP. \quad (7)$$

### 2.2. Internal Penalty Function

When the internal penalty function [38] solves the nonlinear optimization problem, the most important thing is to limit the iteration points in the nonempty feasible domain, which conducts punishment to the feasible points towards the boundary. In addition, the closer the distance towards the feasible domain boundary, the bigger the penalty probability. The purpose of penalty term is to balance the tendency of the feasible domain to be away from the optimal solution when the feasible domain approaches the boundary and the specific form of the internal penalty function is as follows:

$$\Psi_1(x, r_k) = f(x) + r_k B(x), \quad (8)$$

where  $B(x)$  is the penalty term of the internal penalty function, which satisfies the following conditions:  $B(x)$  is continuous inside the feasible domain  $int(D)$ ,  $B(x)$  is a non-negative function. When  $x \rightarrow Round(D)$ ,  $B(x) \rightarrow +\infty$ .  $r_k$  is the internal penalty factor,  $\{r_k\}$  is a monotone decreasing internal penalty factor sequence, which satisfies the following conditions:  $r_k \geq 0$ ,  $\lim_{k \rightarrow +\infty} r_k = 0$ . Let  $x_k$  be an approximate value after the  $k$  iteration, then there are  $\lim_{k \rightarrow +\infty} B(x) = +\infty$  and  $\lim_{k \rightarrow +\infty} \min \Psi_1(x, r_k) = \lim_{k \rightarrow +\infty} \min(f(x) + r_k B(x)) = \min f(x)$ .

#### Normalization of the Internal Penalty Function

Because of the logarithmic function or semi-exponential function converging better in the nonlinear optimization theory, we use the logarithmic function to reduce the interior penalty function term into a formula as follows:

$$B(x) = - \sum_{i=1}^j \ln\left(- \frac{g_i}{\{\min + \max\}_{int(D)} g_i}\right), \quad (9)$$

where  $\{\min + \max\}_{int(D)} g_i$  is the average of the maximum and minimum values within the feasible domain  $int(D)$ , which can balance the searching speed.

### 2.3. External Penalty Function

The main method of the external penalty function in solving the nonlinear optimization problem is to gradually approach the feasible domain boundary from the outside of the feasible domain [39], which will punish the feasible point violating the constraint conditions but will not punish the

minimum point that satisfies the constraint conditions [40]. In addition, its specific form is described as follows:

$$\Psi_2(x, \mu_k) = f(x) + \mu_k P(x), \tag{10}$$

where  $P(x)$  is the penalty term of the external penalty function, which satisfies these conditions as follows:  $P(x)$  is continuous,  $P(x) = 0, \forall x \in \text{int}(D); P(x) > 0, \forall x \notin D$ .  $\mu_k$  is the external penalty factor, which is a monotone increasing positive external penalty factor sequence that satisfies these conditions:  $\mu_k > 0, \lim_{k \rightarrow +\infty} \mu_k = +\infty$ . Let  $x_k$  be an approximate value after the  $k$  iteration, there is  $\lim_{k \rightarrow +\infty} P_k = 0$ , then  $\lim_{k \rightarrow +\infty} \min \Psi_2(x, \mu_k) = \lim_{k \rightarrow +\infty} \min(f(x) + \mu_k P(x)) = \min f(x)$ .

### Normalization of the External Penalty Function

To balance the equality constraint conditions and inequality constraint conditions on the penalty term of the external penalty function, we normalize the specific form of  $P_k$ . In addition, its form is described as follows:

$$P(x) = \sum_{i=1}^j (\max\{\{\min + \max\}_{\text{int}(D)} g_i, 0\})^2 + \sum_{i=j+1}^m (\{\min + \max\}_{\text{Round}(D)} h_i)^2, \tag{11}$$

where  $\{\min + \max\}_{\text{Round}(D)} h_i$  is the average of the maximum and minimum values of the feasible domain boundary  $\text{Round}(D)$ , which can balance the searching speed.

## 3. Differential Evolution Integration Algorithm Based on Mixed Penalty Function Screening Criteria

### 3.1. Mixed Penalty Function

Simply using internal or external penalty function can transform the constraint problem into an unconstrained problem, which reduces the computational difficulty. However, there are two major defects. First, the position of the effective solution under the constraint conditions often has swing effects when the values of the respective penalty factors are different. In other words, if the optimal point lies on the constraint boundary, the optimal point will not be searched by using the two algorithms, but the iterative sequence generated by the two algorithms will be infinitely close to the optimal point. Second, the higher penalty factors are not to easy to be controlled. For example, for the external penalty function, the penalty factor is too large, which causes the iterative points to rotate around the optimal point and be mistaken for the optimal point, which causes incorrect solutions; if the penalty factor is too small, the condition number of matrix of the penalty function term often becomes too large, so that it is difficult for the iterative sequence to converge to the optimal point. To balance the optimal defects of the two methods, we will establish a mixed penalty function as follows:

$$\begin{aligned} \Psi(x, r_k) = & f(x) - Pr_1(k)r_k \sum_{i=1}^j \ln\left(-\frac{g_i}{\{\min + \max\}_{\text{int}(D)} g_i}\right) + Pr_2(k) \frac{1}{r_k} \left(\sum_{i \notin I} (\max\{\{\min + \max\}_{\text{int}(D)} g_i, 0\})^2\right. \\ & \left. + \sum_{i=j+1}^m (\{\min + \max\}_{\text{Round}(D)} h_i)^2\right) \end{aligned} \tag{12}$$

or

$$\Psi(x, r_k) = f(x) + r_k B(x) + \frac{1}{r_k} P(x), \tag{13}$$

where  $\mu_k = \frac{1}{r_k}$ ,  $P(x) = \sum_{i=1}^j (\max\{\{\min + \max\}_{\text{int}(D)} g_i, 0\})^2 + \sum_{i=j+1}^m (\{\min + \max\}_{\text{Round}(D)} h_i)^2 - \sum_{i \notin I} \ln\left(-\frac{g_i}{\{\min + \max\}_{\text{int}(D)} g_i}\right)$ ,  $I = \{x | g_i(x) \leq 0, i = 1, 2, \dots, j\}$ ,  $\{r_k\}$  is a monotone decreasing penalty factor sequence.  $Pr_t, t = 1, 2$  is the probability of fuzzy iterative points (that is, when the penalty

factor reaches the approximate level of the optimal point by iterating, the closer the optimal point is, the bigger the degree of of the iterative point approximating the optimal point. Then, when  $k \rightarrow +\infty$ , the degree of approximation between  $x_k$  and  $x^*$  will be in a fuzzy state), which can adjust different kinds of penalty functions.

### 3.2. The Screening Criteria of the Mixed Penalty Function

When solving nonlinear constrained optimization problems (1), to effectively balance the swing problem generated by the equality penalty and inequality constraints. Let the iterative point be  $x_k$  after  $k$  iterations, and the accuracy of the iterative solution  $x_k$  be a sufficiently small real value  $\varepsilon \in (10^{(-4)}, 10^{(-5)})$ . According to the brief screening rules of the effective solutions proposed by Deb [41], we establish the screening criteria of the mixed penalty function based on the internal penalty function (this screening criterion was established to relax the spatial search area in order to facilitate cross-regional search of the objective function), the external penalty function and their normalization, as follows:

(1) When the two effective solutions  $x_{k_1}$  and  $x_{k_2}$  are inside the feasible domain  $int(D)$  under the constraint conditions, the effective solution with a smaller internal penalty function value is considered the optimal approximate solution and then  $Pr_1 = \frac{x_k}{\sum_{i \in I} x_k}$ ,  $x_k$  satisfies the formula  $f(x_k) = \min(f(x_{k_1}), f(x_{k_2}))$ ;

(2) When the two effective solutions  $x_{k_1}$  and  $x_{k_2}$  appear in the form of certain solutions and are located at the boundary of the feasible domain  $Round(D)$  under the constraint conditions, we take the mean value of the two effective solutions as an optimal approximate solution. Then,  $Pr_1 = \frac{\min\{x_{k_1}, x_{k_2}\}}{\sum_{i \in I \cup i \notin D} x_k}$ ,  $Pr_2 = \frac{\max\{x_{k_1}, x_{k_2}\}}{\sum_{i \in I \cup i \notin D} x_k}$ ;

(3) When the two effective solutions  $x_{k_1}$  and  $x_{k_2}$  appearing in the form of uncertain solutions are located at the boundary of the feasible domain  $Round(D)$  under the constraint conditions, let  $x_{k_1}$  be an internal iterative solution, then all internal iterative solutions constitute a positive monotonic decreasing sequence  $\{x_{k_1}\}$  within the feasible domain. In addition, let  $x_{k_2}$  be an external iterative solution, then all external iterative solutions constitute a positive monotonic increasing sequence outside the feasible domain. At the same time, let  $x_{k_1} = \inf\{x_{k_1}\}$ ,  $x_{k_2} = \sup\{x_{k_2}\}$ , then we take the mean value of the two effective solutions  $\frac{\inf\{x_{k_1}\} + \sup\{x_{k_2}\}}{2}$  as an optimal approximate solution. Further, there is the following:  $Pr_1 = \frac{\max\{\inf\{x_{k_1}\}, \sup\{x_{k_2}\}\}}{\sum_{i \in int(I) \cup i \notin I} x_k}$ ,  $Pr_2 = \frac{\max\{\inf\{x_{k_1}\}, \sup\{x_{k_2}\}\}}{\sum_{i \in int(I) \cup i \notin I} x_k}$ ;

(4) When the two effective solutions  $x_{k_1}$  and  $x_{k_2}$  are outside the feasible domain under constraint conditions, the effective solution with the larger exterior penalty function value will be considered as the optimal approximate solution and then  $Pr_2 = \frac{x_k}{\sum_{i \in I} x_k}$ ,  $x_k$  will satisfy the formula  $f(x_k) = \max(f(x_{k_1}), f(x_{k_2}))$ ;

(5) When the solution of the constraint is an invalid solution, in order to ensure the global nature of the algorithm search, we will perform the domain expansion operation based on the probability of the two invalid solutions: Firstly, the invalid solution  $x'_1$  is taken as the center and the original space search area is dimension expanded with the preset precision  $\varepsilon \in (-1, 1)$  as the radius, so that the constraint condition is relaxed and the invalid solution is validated.

The screening criterion possesses the following properties:

1. The internal penalty function or the external penalty function can only detect the inside of the feasible domain or the outside of the feasible domain; they are unable to detect the feasibility of the area by crossing the regional boundary, which can divide the feasible domain into two parts, interior and exterior. However, by adding the probability condition based on the penalty function, we can balance the approximate optimal solution  $x_k$  according to the approximate degree of the iterative points  $x_k$  and the global optimal point  $x^*$ . In addition, the criterion (2) or (3) plays a balancing role in the optimal conditions, avoiding the phenomenon of the solution that cannot move in the interior and exterior of the feasible domain.

2. When the iterative points  $x_k$  completely fall in the interior  $int(D)$  or exterior  $out(D)$  of the feasible domain, the criteria (1) or (4) has the characteristics of being able to flexibly select the optimal area, which cannot only avoid the loss of the approximate global optimal point  $x^*$  caused by a single optimal area but also ensure searching the iterative points in the whole area. Further, it can avoid the occurrence of an approximate solution with less precision due to the singularity of the solution.

3. The criterion (1) or (4) adopts the selection strategy in probability, which has better optimal effect for the strong constraint optimization problem [42]. This is essentially an intensive strategy for elite retention strategies, so it can be applied to stronger constraint optimization problems.

4. The internal penalty function tends to select the iterative points that are far from the feasible domain boundary and the external penalty function tends to select the iterative points that are closer to the feasible domain boundary  $Round(D)$ , both of which are not conducive to the global convergence of the optimal point. However, the criterion can effectively avoid the dispersion distribution of iterative points and can find the global optimal point faster in the condition of the induction of the differential evolution algorithm.

### 3.3. The Processing of the Constraint Conditions

In the constraint optimization problems shown by (1), there are two kinds of constraints: equality constraints and inequality constraints. Inequality constraints tend to expand the searching range, which is beneficial to global searching and can reduce the probability of fault tolerance. However, The equality constraints present a one-dimensional linear region, which can narrow the optimal range and is not conducive to global optimization. For this reason, we conduct high-dimension processing on the equality constraints:

$$g_i = |h_i| - \varepsilon_i \leq 0, \tag{14}$$

where  $\varepsilon_i \geq 0$  is a sufficiently small real-valued slack variable, which increases the searching range by expanding the dimension, which is beneficial to global searching and global convergence. However, the phenomenon will reduce the feasible domain searching ratio, which is not beneficial to finding the global convergent point  $x^*$ . To solve the problem, we adopt the method of the self-adaptive slack variable to deal with the equality constraints [43].

$$If(R_n(k) \leq R_l), Then \varepsilon_i(t + 1) = \beta_l \varepsilon_i(k) \tag{15}$$

$$If(R_n(k) \geq R_u), Then \varepsilon_i(t + 1) = \beta_u \varepsilon_i(k), \tag{16}$$

where  $R_n$  is the proportion of feasible solutions satisfying the slack variable in the current population,  $0 \leq R_l \leq R_u \leq 1, 0 \leq \beta_l \leq 1 \leq \beta_u, \varepsilon_i(0)$  is the maximum value that violates the effective solution in the initial population.

### 3.4. Implementation of Differential Evolution Integrated Algorithm Based on the Screening Criterion of the Mixed Penalty Function

In this paper, the mixed penalty function screening criterion is applied to the DE algorithm, the validity and the convergence of the algorithm are analyzed by the Markov model. Finally, we obtain the differential evolution integrated algorithm based on mixed function screening criteria (DE-MPFSC algorithm).

In the DE-MPFSC algorithm, let the space dimension be  $D = n$ , the population individuals are a set of  $S_X = \{(x_i^t, \delta_i) | i = 1, 2, \dots, NP\}$ ,  $x_i^t$  is an  $n$ -dim decision variable,  $\delta_i$  is the  $n$ -dim individual iterative step variable. The initial population is evenly distributed in the  $n$ -dim space constraints. The mutation operator, crossover operator and selection operator  $DE$  algorithm can act on the  $n$ -dim

space constraints and work on all individuals in the initial population to improve their searching ability, where the searching step calculation formula of the DE-MPFSC algorithm is as follows:

$$\delta_i^{t+1} = \delta_i^t + \exp(\tau \cdot N(0, 1) + \tau' \cdot N_i(0, 1)) \quad (17)$$

$$x_i^{t+1} = x_i^t + \delta_i^{t+1} \cdot N_i(0, 1), \quad (18)$$

where  $t$  is the number of iterations, and  $\tau$  and  $\tau'$  are the self-adaptive learning rate of the population individuals. To ensure the accuracy of the calculation, we calculate according to the method proposed by Schwefel [44]:  $\tau = (\sqrt{2\sqrt{n}})^{-1}$  and  $\tau' = (\sqrt{2n})^{-1}$ .  $N(0, 1)$  and  $N_i(0, 1)$  are all real uniform Gaussian distributions with a mean of 0 and a variance of 1. Then, we select individuals in the algorithm according to the mixed penalty function screening criteria and keep the outstanding individuals of  $\zeta$  parent individuals and  $\xi$  progeny individuals to the next generation. The DE-MPFSC algorithm steps are as follows:

**STEP 1:** Initializing variables: According to the number of the population and the number of individuals, let  $t = 0$ , then generate the initial population within  $\psi$  parents individuals and  $\varphi$  progeny individuals and each of the individuals corresponds with  $\{(x_i^t, \delta_i) | i = 1, 2, \dots, NP\}$ .

**STEP 2:** Calculating the fitness function value of the initial variable: Calculate the fitness function values of  $\psi$  parent individuals and record the maximum value and minimum value. At the same time, calculate the fitness function values at each iterative points inside and outside the feasible domain under the condition of the initial conditions and records the maximum value and minimum value.

**STEP 3:** According to Formulas (5), (17) and (18), generate corresponding  $\psi$  mutation individuals.

**STEP 4:** Calculating the fitness function value of corresponding  $\psi$  mutation individuals and calculating the probability of the fitness function value of the iterative points of different regions based on screening criterion of the mixed penalty function.

**STEP 5:** Conducting selection and crossover operations for corresponding  $\psi$  mutation individuals according to Formulas (6) and (7), generating  $\varphi$  progeny individuals and calculating their fitness function values and recording the iterative points that accord with the accuracy of the problem.

**STEP 6:** Consisting of the  $\psi$  mutation individuals and the  $\varphi$  progeny individuals generated by the selection and crossover operations into a new  $\psi + \varphi$  individuals. According to the screening criteria, choose  $\psi$  individuals of  $t$  generation as  $t + 1$  generation parents individuals and record the iterative points  $x_{k+1}$  that accord with the accuracy of the problem.

**STEP 7:** Generating a series of iterative points  $\{x_{k_1}\}, \{x_{k_2}\}$  in two parts inside and outside the feasible domain.

**STEP 8:** According to the iterative points sequence  $\{x_{k_1}\}, \{x_{k_2}\}$  generated by step 7, the former are arranged in a monotonic decreasing sequence and the latter is arranged in a monotonic increasing sequence.

**STEP 9:** Calculating  $Pr_1, Pr_2$  according to the probability formula of the screening criteria.

**STEP 10:** Substituting the result of step 9 into Formulas (12) or (13), calculating the fitness function values and iterative points and judging whether the accuracy is satisfied.

**STEP 11:** If the iterative points satisfy the screening criteria and accuracy, then terminate the algorithm flows, otherwise let  $t = t + 1$ , return step 3.

#### 4. Theoretical Analysis of DE-MPFSC Algorithm

Karmer [45] pointed out through experiments and analysis that, when solving the nonlinear optimization problem with constraints, if the optimal solution is at the feasible domain boundary, using a single objective function as the test function is not conducive to searching the optimal point, which easily falls into local optimization, resulting in an error solution. To effectively avoid the phenomenon, this paper uses a selection variable with adaptive accuracy to modify the equality constraints, expands the feasible domain searching range, establishes the DE-MPFSC algorithm. First, we theoretically analyze the validity of the DE-MPFSC algorithm (judging whether the Markov

model conditions of the DE-MPFSC algorithm is true) and the convergence analysis (judging whether the DE-MPFSC algorithm possesses local or global convergence).

#### 4.1. Validity Analysis of DE-MPFSC Algorithm

The validity analysis of the DE-MPFSC algorithm is that whether the Markov model condition of the DE-MPFSC algorithm is true in the constrained feasible domain. It is necessary to judge whether the state transition of the population sequence is a Markov chain. So, we introduce the corresponding definitions and lemmas.

**Definition 1** ([46,47]). Let  $\{\hat{X}_n; n \geq 0\}$  be a random variable with a discrete value. The whole discrete values are recorded as  $S$ , which is called the state space. If  $\forall n \geq 1, i_k \in S(k \leq n + 1)$  and  $P\{\hat{X}_{n+1} = \frac{i_{n+1}}{\hat{X}_n} = i_n, \dots, \hat{X}_0 = i_0\} = P\{\hat{X}_{n+1} = \frac{i_{n+1}}{\hat{X}_n} = i_n\}$ , then  $\{\hat{X}_n; n \geq 0\}$  is called the Markov chain.

**Lemma 1** ([46,47]). The joint distribution of homogeneous Markov chain  $p_{ij}^n = P\{\hat{X}_{n+m} = \frac{j}{\hat{X}_m} = i\}$  is determined by the initial distribution  $P\{\hat{X}_0 = i\} = p_i, i \in S$  and the individual transition probability  $p_{ij} = P\{\hat{X}_n = \frac{j}{\hat{X}_{n-1}} = i\}, (i, j \in S)$ .

**Lemma 2** ([46,47]). For the homogeneous Markov chain  $p_{ij}^n = P\{\hat{X}_{n+m} = \frac{j}{\hat{X}_m} = i\}$  and  $\forall n, m \geq 0, i, j \in S$ , there are  $p_{ij}^{n+m} = \sum_{k \in S} p_{ik}^n p_{kj}^m$ .

**Definition 2** ([46,47]). Let  $\{\hat{X}_n; n \geq 0\}$  be the finite homogeneous Markov chain,  $p_{ij}^n$  is the  $n$ th iterative individual transition probability. If  $\exists n \geq 1$  makes  $p_{ij}^n \geq 0$ , which is called a case in which status  $i$  can be transferred to state  $j$ ; otherwise, is called another case in which the status  $i$  cannot be transferred to state  $j$ .

**Definition 3** ([46,47]). For Markov chain  $\{\hat{X}_n; n \geq 0\}$ , the greatest common divisor  $d_i$  of the state set  $\{i | n \geq 1, p_{ij}^n \geq 0\}$  is called the generalized period of the state set  $i$ . If  $d_i > 1$ , then state set  $i$  is periodical. If  $d_i = 1$ , then state set  $i$  is nonperiodic. If  $d_i$  is a nonpositive real number, then  $i$  cannot be periodical.

**Lemma 3** ([46,47]). If Markov chain  $\{\hat{X}_n; n \geq 0\}$  is irreducible, that is, that all individual states are connected to each other, and  $\exists j \leq N$ , and  $\exists j \leq N$  make  $p_{ij} > 0$ , then the chain is nonperiodic. Its transfer matrix is the primitive random matrix and there is a stationary distribution on the transfer matrix. Where there is a stationary distribution and a limited distribution  $\lim_{n \rightarrow \infty} P\{X_n = j\} = o_j$  on the transfer random matrix of the Markov chain being nonperiodic, irreducible and a finite state.

Now, consider the Markov processing model  $\{(x_i^t, \delta_i) | i = 1, 2, \dots, NP\}$ , the progeny  $x_k^{t+1}$  and iterative steps  $\delta_i$  of the DE-MPFSC algorithm are determined by the following formula.

$$\delta_i^{t+1} = \begin{cases} \gamma \delta_i^t, & \text{If } \Psi(x_k^t + \delta_i^t C^t) < \Psi(x_k^t), \\ & \text{and } g_i(x_k^t + \delta_i^t C^t) \leq 0 \\ \gamma^{-1} \delta_i^t, & \text{otherwise} \end{cases} \tag{19}$$

$$x_k^{t+1} = \begin{cases} x_k^t + \delta_i^t C^t, & \text{If } \Psi(x_k^t + \delta_i^t C^t) < \Psi(x_k^t), \\ & \text{and } g_i(x_k^t + \delta_i^t C^t) \leq 0 \\ x_k^t, & \text{otherwise} \end{cases} , \tag{20}$$

where  $\gamma > 1$  is a compilation parameter,  $\{C^t, t \geq 0\}$  is a random vector and independent and identically distributed.

Assume that the new individuals generated by the mutation operation of the DE algorithm are distributed on a circle centered on  $x_k^t$  with a radius of  $\delta_i^t$ . When the mutation individual generated by the DE-MPFSC algorithm is valid, the step increases; otherwise, the step decreases.

Where effective mutation individuals are superior than the parents. Then, each evolution can be performed simultaneously from the inside and outside of the feasible domain.

Markov Processing Model of DE-MPFSC Algorithm

Assume the individuals in the population are expressed as  $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$ ,  $i = 1, 2, \dots, NP$  [20], where  $t$  is the number of iterations,  $NP$  is population size [20], the mutation operator, crossover operator, and selection operator are, respectively,  $F, T_c, T_s, T = F \cdot T_c \cdot T_s$ . Then, the iterative equation of the DE-MPFSC algorithm is  $X(t + 1) = T(X(t)) = (F \cdot T_c \cdot T_s)(X(t))$ . We obtain the population sequence  $\{\hat{X}_t; t \geq 0\}$  after initializing the population.

**Theorem 1.** Let the population sequence of the DE-MPFSC algorithm be  $\{\hat{X}_t; n \geq 0\}$ ; then, the sequence is a finite, homogeneous, irreducible and nonperiodic Markov chain.

**Proof.** Let  $X(t)$  be the population of the DE-MPFSC algorithm, the population individuals can be  $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t), i = 1, 2, \dots, NP$ .  $S = \{\hat{X}_n; n \geq 0\}$  is the state space of the population sequence, which is the finite space. Because of  $X(t + 1) = T(X(t)) = (F \cdot T_c \cdot T_s)(X(t))$ , where there is no connection between operators  $F, T_c, T_s, T = F \cdot T_c \cdot T_s$  of the DE-MPFSC algorithm and the iterative times  $t$ , and  $X(t + 1)$  is only related to  $X(t)$ . In other words, the evolved population individual is only related to the corresponding individual before the evolution, not the number of evolutions. Therefore,  $\{\hat{X}_t; t \geq 0\}$  is a finite state Markov chain. Due to

$$\begin{aligned}
 &P\{T(\hat{X}_t)_k = X_k(t + 1)\} \\
 &= \sum_{X'_k(t+1) \in S} \sum_{(X_1^k(t), X_2^k(t)) \in S} P\{T_s(X(t))(X_1^k(t), X_2^k(t))\} \cdot P\{T_c(X_1^k(t), X_2^k(t))X'_k(t + 1)\} \cdot P\{F(X'_k(t + 1))\} \\
 &= X_k(t + 1),
 \end{aligned} \tag{21}$$

where  $X'_k(t + 1)$  is a population of individuals generated by a crossover operation, and  $\forall X(t) \in S$ . Then, there is the case in which  $(X_1^k(t), X_2^k(t))$  and  $X'_k(t + 1)$  make

$$P\{T_s(X(t)) = (X_1^k(t), X_2^k(t))\} > 0 \tag{22}$$

$$P\{T_c(X_1^k(t), X_2^k(t)) = X'_k(t + 1)\} > 0. \tag{23}$$

Further, for  $\forall X'_k(t + 1)$  and  $\forall X_k(t + 1)$ , there is

$$P\{F(X'_k(t + 1)) = X_k(t + 1)\} > 0 \tag{24}$$

Then,

$$P\{T(\hat{X}_t)_k = X_k(t + 1)\} > 0. \tag{25}$$

Based on formula (22), (23), (24), we know that formula (25) is connected with  $n$ , then

$$P\{T(X(t)) = X(t + 1)\} = \prod_{i=1}^{NP} P\{T(\hat{X}_t)_k = X_k(t + 1)\} > 0 \tag{26}$$

has no connection with  $n$ .  $\square$

Relying on these proving processes, we know that the population sequence  $\{\hat{X}_t; n \geq 0\}$  generated by the DE-MPFSC algorithm is a homogeneous, irreducible and nonperiodic Markov chain. The conclusion is true.

4.2. Convergence Analysis of DE-MPFSC Algorithm

**Definition 4** ([48–50]).  $\{\hat{X}_n; n \geq 0\}$  is the set that is strongly convergent to the global optimal solution in probability. If  $\lim_{n \rightarrow \infty} P\{\hat{X}_n \subset M\} = 1$  is true, which is marked as  $\hat{X}_n \rightarrow M(P.S.)$ .

**Definition 5** ([48–50]).  $B \subset S$  is a satisfactory solution set. If  $\forall X \in B, Y \notin B$  make  $f(X) > f(Y)$ .

**Theorem 2.** (Theorem 2 shows that the Markov chain of the DE-MPFSC algorithm is convergent in the distribution and does not depend on the choice of the initial population, which is expressed as  $o(Z) = \sum_{Y \in S} P\{X_t = \frac{Z}{X_0} = Y\}$ ). (Limit Distribution Theorem) There is a limited distribution for the population sequence  $\{\hat{X}_n; n \geq 0\}$  of the DE-MPFSC algorithm.

**Proof.** From Theorem 1, we know that the sequence  $\{\hat{X}_t; n \geq 0\}$  of the DE-MPFSC algorithm is a finite, homogeneous, irreducible and nonperiodic Markov chain. Then, from lemma 3, we obtain

$$\lim_{t \rightarrow \infty} P\{T(X_t) = Y\} = \lim_{t \rightarrow \infty} \sum_{X_0 \in S} P\{T(X_t) = \frac{Y}{X_0}\} P\{X_0\} = o(Y). \tag{27}$$

Then,  $o(Y)$  is the distribution on state population  $S$  and

$$\sum_{Y \in S} P\{X_t = \frac{Z}{X_0} = Y\} = o(Z), (Z \in S). \tag{28}$$

□

**Theorem 3.** (The Strong Convergence Theorem in the Probability of the DE-MPFSC Algorithm) (Theorem 3 shows that the convergence of the Markov sequence  $\{X_t\}, \{X_t\}$  on the state population  $S$  of the DE-MPFSC algorithm is related to parameters  $\{\alpha_t^D\}, \{\beta_t^D\}$ .  $\alpha_t^D$  is the probability that the population individuals in a Markov sequence  $\{X_t\}$  will leave the next satisfactory solution sets [50]. However, from Formulas (1), (2) of theorem 3,  $\alpha_t^D$  is a negative correlation with  $\beta_t^D$ . So, for the Markov sequence  $\{X_t\}$ , there is a nonconvergent state. In order to enhance the convergence analysis of the satisfactory solution to the DE-MPFSC algorithm, we need to perform strong convergence analysis around all feasible points that satisfy the limit distribution. The theorem theoretically explains the necessary conditions for the absolute convergence of the DE-MPFSC algorithm.). Let  $P$  be the probability distribution on the state population  $S$ ,  $\{X(t)\}$  is the Markov sequence on  $S$ ,  $M$  is the global optimal solution set,  $D$  is any satisfactory solution set. If  $\{\tilde{\alpha}_t^D\}, \{\tilde{\beta}_t^D\}$  satisfies these following conditions: (1)  $\sum_{t=1}^{\infty} (1 - \tilde{\beta}_t^D) = \infty$ ; (2)  $\lim_{t \rightarrow \infty} \frac{\tilde{\alpha}_t^D}{1 - \tilde{\beta}_t^D} = 0$ .

Then,  $X_t$  is strongly convergent in probability to satisfactory solution sets  $D$ , which is to say  $\lim_{t \rightarrow \infty} P\{X_t \subset D\} = 1$ . Where  $\tilde{\alpha}^D = P\{X(t+1) \cap D^c \neq \emptyset\}, \tilde{\beta}^D = P\{X(t+1) \cap D^c = \emptyset\}$ .

**Proof.** Let  $P_0 = P\{X_t \cap D = \emptyset\}$ , from the Bayes formula, we know

$$\begin{aligned} P_0(t+1) &= P\{X(t) \cap D = \emptyset\} \\ &= P\{X(t+1) \cap D \neq \emptyset\} \cdot P\{X(t) \cap D \neq \emptyset\} + P\{X(t+1) \cap D = \emptyset\} \cdot P\{X(t) \cap D = \emptyset\} \\ &\leq \tilde{\alpha}_t^D + \tilde{\beta}_t^D P_0(t). \end{aligned} \tag{29}$$

From Formula (4), we know  $\forall \varepsilon > 0, \exists N_1$ , when  $t \geq N_1, \frac{\tilde{\alpha}_t^D}{1 - \tilde{\beta}_t^D} \leq \frac{\varepsilon}{2}$ ; thus,  $\tilde{\alpha}_t^D \leq \frac{\varepsilon}{2}(1 - \tilde{\beta}_t^D)$ . From Formula (29), we obtain

$$(P_0(t+1) - \frac{\varepsilon}{2}) - \tilde{\beta}_t^D (P_0(t) - \frac{\varepsilon}{2}) \leq P_0(t+1) - \tilde{\beta}_t^D P_0(t) - \tilde{\alpha}_t^D \leq 0. \tag{30}$$

Further, there is  $P_0(t + 1) - \frac{\varepsilon}{2} \leq \tilde{\beta}_t^D (P_0(t) - \frac{\varepsilon}{2})$ , then we know

$$(P_0(t + 1) - \frac{\varepsilon}{2}) \leq \frac{\varepsilon}{2} + \prod_{k=1}^t \tilde{\beta}_k^D (P_0(t) - \frac{\varepsilon}{2}) \tag{31}$$

Additionally,  $\lim_{t \rightarrow \infty} \frac{\tilde{\alpha}_t^D}{1 - \tilde{\beta}_t^D} \Leftrightarrow \prod_{k=1}^t \tilde{\beta}_k^D = 0$ . Then, when  $\exists N_2, t \geq N_2$ , there is  $\prod_{k=1}^t \tilde{\beta}_k^D \leq \frac{\varepsilon}{2}$ . Further, when  $t \geq \max\{N_1, N_2\}$ , there is  $P_0(t + 1) \leq \varepsilon$ . Therefore, we obtain

$$\lim_{t \rightarrow \infty} P\{X(t) \cap D = \emptyset\} = \lim_{t \rightarrow \infty} P_0(t) = 0. \tag{32}$$

Further, there is  $\lim_{t \rightarrow \infty} P\{X_t \cap D \neq \emptyset\} = 1$ . This shows that DE-MPFSC is strongly convergent.  $\square$

### 5. Entanglement and Numerical Test of DE-MPFSC Algorithm Stability

We focus on the effect of the DE-MPFSC algorithm in imbalanced data integration, rather than focusing on the advantages of this algorithm over other algorithms. To test the advantages of the DE-MPFSC in processing the imbalanced data and to ensure the high property of the DE-MPFSC algorithm in the imbalanced data integration process compared to the traditional DE algorithm, we will conduct test experiments on the stability and effectiveness of the DE-MPFSC algorithm. Therefore, based on the topology structure between the convergence rate and convergence precision of the individuals and the quantum relationship [20], including the entanglement phenomenon of individual convergence speed and convergence precision in the evolution process, we introduce the concept of entanglement degree  $\frac{1}{\xi}$  and the entanglement degree error  $\zeta$ , analyze the stability of DE-MPFSC algorithm and DE algorithm from system theory.

#### 5.1. Entanglement Degree $\frac{1}{\xi}$ and Entanglement Degree Error $\zeta$

**Lemma 4** ([20]). Let  $f_\varepsilon$  be a continuous differentiable function defined in a complete normed linear space and  $f_\varepsilon(v_1, \dots, v_n) \in L_2(\mathbb{R}_n), M \in Sp(2n + P_t, \mathbb{R}), M^n$  is a complete normed linear space.  $\{S_i^n | i = 1, 2, \dots, n\}$  is a generalized  $n$  dimension complete normed linear subspace and  $M^n = S_1^n \cup S_2^n \cup \dots \cup S_n^n$  forms an open cover of  $M^n$ , then  $\forall \lambda_i^t \in \{\lambda_i^t | i = 1, 2, \dots, n\}, \exists S_i^n, \lim_{t \rightarrow \infty} \lambda_i^t \sim \lambda_i$  or  $\lim_{t \rightarrow \infty} \lambda_i^t = (\lambda_\varepsilon)_i, \lambda_i \in S_i^n$ . If  $\det(B) \neq 0$ , then the following formula,

$$\begin{aligned} \Delta_v^2 \cdot \Delta_{x_\beta^\varepsilon}^2 &\geq \left( \frac{\sqrt{\lim_{t \rightarrow \infty} \lambda_1^t} + \dots + \sqrt{\lim_{t \rightarrow \infty} \lambda_n^t}}{2} \right)^2 \\ &= \left( \frac{\sqrt{(\lambda_\varepsilon)_1} + \dots + \sqrt{(\lambda_\varepsilon)_n}}{2} \right)^2. \end{aligned} \tag{33}$$

The lemma is detailed in Reference [20], which theoretically shows that the convergence accuracy and convergence speed of the evolutionary individual in the iterative process represent an uncertain quantum relationship. The lemma is important for analyzing the distribution of the optimal point of the segmentation regions. The feasible points distributed in different regions force the internal and external penalty functions to search in different segmentation regions by the jumping function of probability. Let the region have two local optimal feasible points  $x_1$  and  $x_2$ , which are distributed to satisfy  $\Delta_v^2 \cdot \Delta_{x_\beta^\varepsilon}^2 \geq \left(\frac{\sqrt{x_1} + \sqrt{x_2}}{2}\right)^2$  and  $\sqrt{\lim_{t \rightarrow \infty} x_1^t} = \sqrt{x_1}, \sqrt{\lim_{t \rightarrow \infty} x_2^t} = \sqrt{x_2}$ , where  $t$  is the number of individual iterations.

The individual entanglement degree we define is  $\frac{1}{\xi} = \frac{v_{X(t)}}{t_{X(t)}}$ , which is the numerical ratio between the convergence speed and convergence precision of individuals during population evolution. The higher the entanglement degree, the better the convergence stability of the algorithm or the faster convergence of the algorithm and the higher convergence precision. The entanglement degree error  $\zeta$  is the correlation of the entanglement effect, which is the efficiency of the entanglement decided

by  $\frac{1}{\zeta} \ln Cov(NP) = \zeta \ln E(NP - X_i)$ . The smaller the entanglement degree error, the smaller the fluctuation of the algorithm in the iterative process. Where,  $v_{X(t)}$  is the individual convergence speed,  $\iota(X(t))$  is the individual convergence precision,  $\frac{\ln Cov(NP)}{\ln E(NP - X_i)}$  is the entanglement coefficient,  $\zeta$  is the discrete degree between entangled individuals. About the entangled relationship between convergence precision and convergence speed, we can refer to the topological geometric relationship described by the author of Reference [20]. The effect of entanglement degree clearly shows the efficiency of algorithm convergence in the geometry.

5.2. Entangled Image and Analysis of DE-MPFSC Algorithm

Since data integration is closely related to data size and data dimension, we first reduces dimensionality of high-dimensional data before performing data experiments. Let the number of population individuals is 1000. The experiment is divided into 5 times on average, and the number of individuals per test is gradually increased from 200. Test runs an average of 5 times. The initial variation factor and initial crossover probability of the algorithm are determined by the literature [19,20]. The initial mutation factor and the initial crossover probability are respectively set to  $F = 0.2, CR = 0.5$ . In the following experiments,  $F = 0.2, CR = 0.5$  are dynamically adjusted according to the DE-MPFSC algorithm. The Figures 1–10 below shows the numerical image of the DE-MPFSC algorithm.

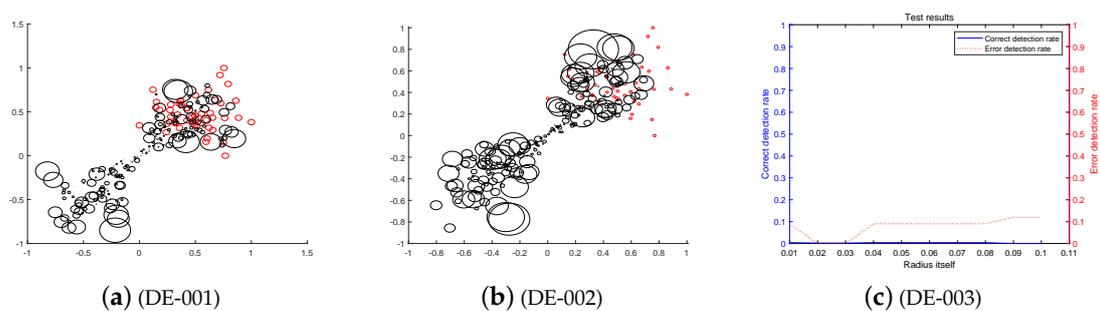


Figure 1. The entangled image and error of the differential evolution (DE) algorithm running 200 times.

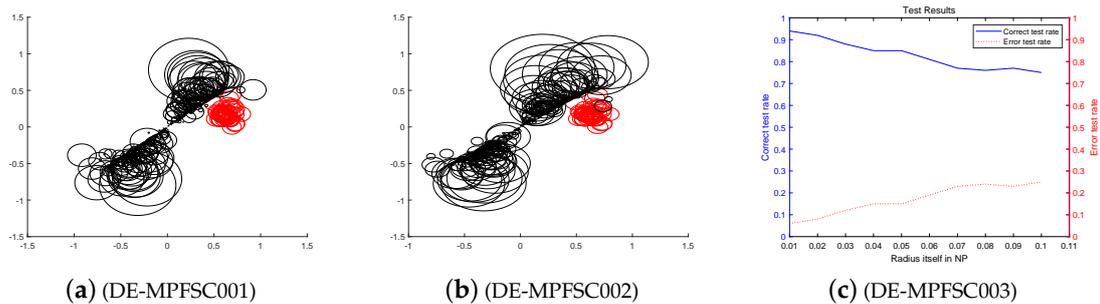


Figure 2. The entangled image and error of the Differential Evolution Integration Algorithm Based on Mixed Penalty Function Screening Criteria (DE-MPFSC) algorithm running 200 times.

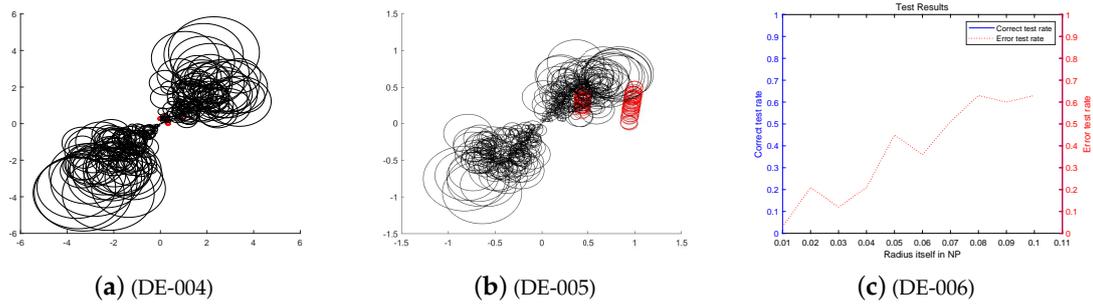


Figure 3. The entangled image and error of the DE algorithm running 400 times.

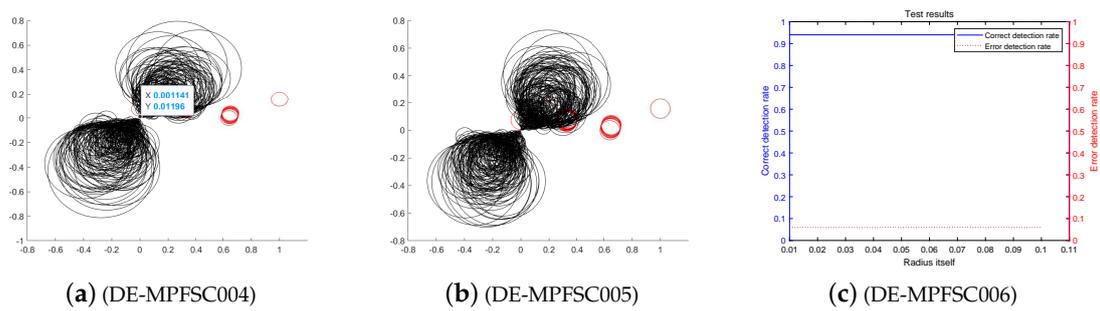


Figure 4. The entangled image and error of the DE-MPFSC algorithm running 400 times.

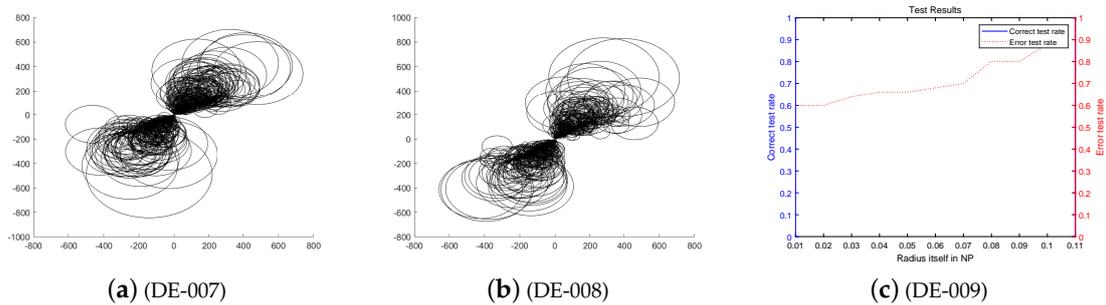


Figure 5. The entangled image and error of the DE algorithm running 600 times.

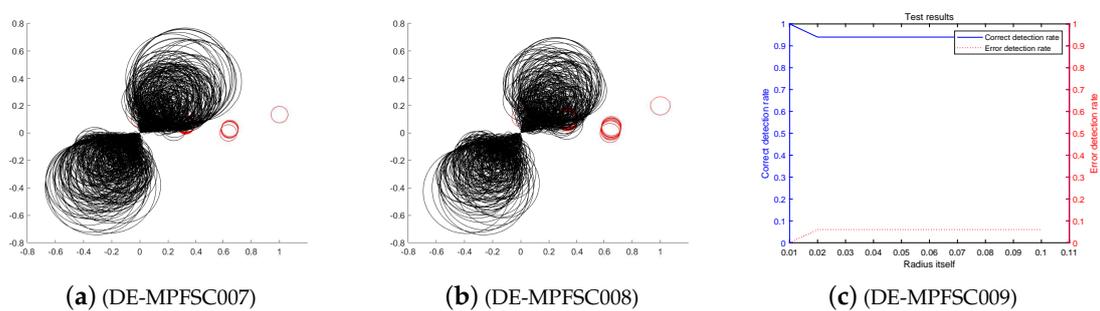


Figure 6. The entangled image and error of the DE-MPFSC algorithm running 600 times.

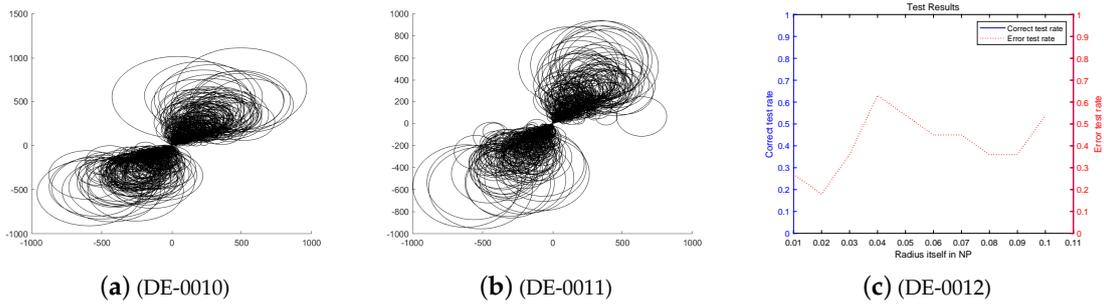


Figure 7. The entangled image and error of the DE algorithm running 800 times.

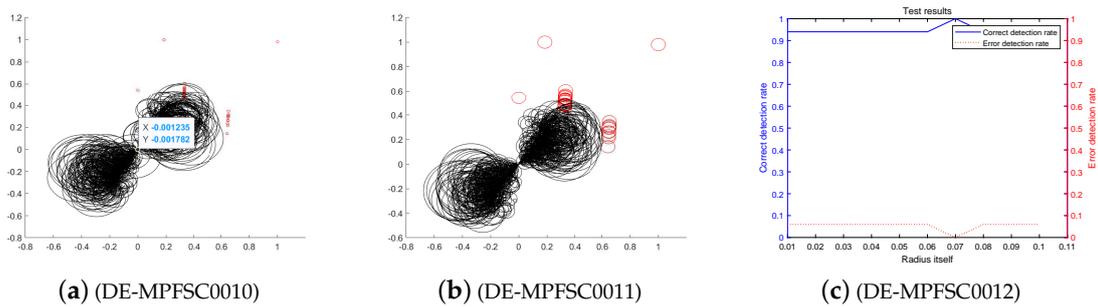


Figure 8. The entangled image and error of the DE-MPFSC algorithm running 800 times.

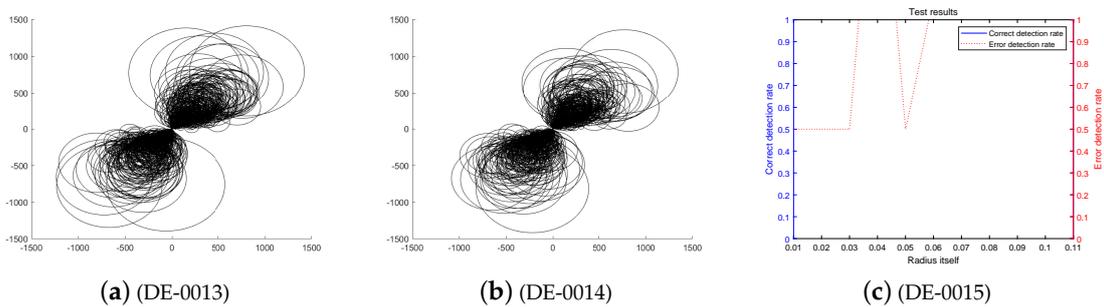


Figure 9. The entangled image and error of the DE algorithm running 1000 times.

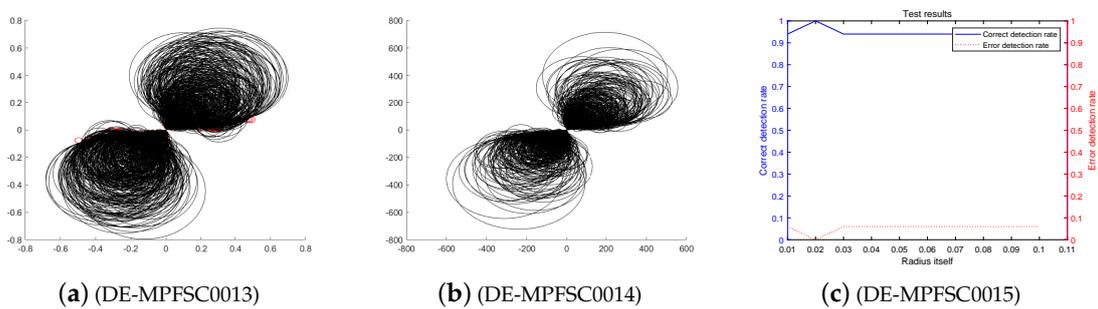


Figure 10. The entangled image and error of the DE-MPFSC algorithm running 1000 times.

Figures 1–10 above present entanglement degree analysis about the stability and validity between convergence precision and convergence speed of population individuals.

First, we analyze the stability and effectiveness of the DE-MPFSC algorithm from the number of population individuals. When the number of individuals are 200, 400, 600, 800, 1000, the DE algorithm entanglement degree about convergence precision and convergence speed are divided into two branches. However, the convergence speed and convergence precision are higher discrete and they

do not converge to the entanglement center point along the convergence curve. The error test rates are 0.1, 0.65, 0.65, 0.55, 1, the correct test rate is 0 and the detection efficiency is 0, which shows that when the population individuals are 200, 400, 600, 800, 1000, the stability of the DE algorithm is not high and it is easy to generate fluctuations. Compared with the DE algorithm, the entanglement degree of the DE-MPFSC algorithm about convergence precision and convergence speed are divided into two branches. The convergence speed and convergence precision are lower discrete and they converge to the entanglement center point along the convergence curve. The error test rates are 0.25, 0.05, 0.05, 0.05, 0.05, the correct test rate are 0.75, 0.95, 0.95, 0.95, 0.95 and the detection efficiency are 3, 19, 19, 19, 19, which shows that when the population individuals are 200, 400, 600, 800, 1000, the stability of the DE-MPFSC algorithm is higher than DE algorithm.

In particular, when the number of individuals is 1000, the convergence speed and convergence precision are slightly lower than that of 800 individuals but the distribution is still scattered. They generate a more dramatic marginal differentiation phenomenon on branches and they do not converge to the entanglement center point along the convergence curve. The error test rate is 1, the correct test rate is 0 and the detection efficiency is 0. The situation must not be used as a basis for effective convergence, which shows that when the population individuals is 1000, the stability of the DE algorithm cannot be accepted and it is easy to generate fluctuations. Compared with the DE algorithm, the entanglement degree of the DE-MPFSC algorithm is divided into two branches. The convergence speed and convergence precision are lower discrete, which can further weak marginal differentiation phenomenon on branches and they converge to the entanglement center point along the convergence curve.

In summary, the stability of the DE-MPFSC algorithm is higher than that of the DE algorithm.

### 5.3. Numerical Test and Analysis of DE-MPFSC Algorithm

Since the data integration is closely related to the data size and data dimension, we set the population individuals to 1000, which is divided into 5 experiments and the average operation is 5 times for each experiment. According to the recommendations of Professor Storn R and Professor Price in Reference [19], the initial mutation factor and initial crossover probability are set to  $F = 0.2, CR = 0.5$ . Tables 1–5 below shows the numerical effects of (Differential Evolution Algorithm Based on Mixed Penalty Function Screening Criteria)DE-MPFSC algorithm and (Differential Evolution)DE algorithm.

**Table 1.** Numerical Analysis of Entanglement Degree  $\frac{1}{\xi}$  and Entanglement Degree Error  $\zeta$  of the Differential Evolution (DE) algorithm and the Mixed Penalty Function Screening Criteria (DE-MPFSC) algorithm 200 times.

NP	DE					DE – MPFSC				
	F	CR	$\xi$	$\zeta$	$\epsilon\%$	F*	CR*	$\xi$	$\zeta$	$\epsilon\%$
200	0.1	0.9	0.8	0.5	$\pm 0.625$	0.1	0.9	0.12	0.05	$\pm 0.416$
200	/	/	0.73	0.66	$\pm 0.904$	0.5	0.6	0.20	0.06	$\pm 0.3$
200	/	/	0.81	0.49	$\pm 0.604$	0.3	0.2	0.19	0.11	$\pm 0.578$
200	/	/	0.75	0.54	$\pm 0.72$	0.2	0.3	0.15	0.03	$\pm 0.2$
200	/	/	0.8	0.56	$\pm 0.7$	0.2	0.3	0.15	0.02	$\pm 0.13$
Average	DE					DE – MPFSC				
	0.1	0.9	0.778	0.55	$\pm 0.7106$	0.26	0.46	0.162	0.054	$\pm 0.3248$
Variance	DE					DE – MPFSC				
	0	0	0.00127	0.0046	$\pm 0.01407$	0.023	0.083	0.00107	0.00123	$\pm 0.03164$

**Table 2.** Numerical Analysis of Entanglement Degree  $\frac{1}{\xi}$  and Entanglement Degree Error  $\zeta$  of the DE algorithm and the DE-MPFSC algorithm 400 times.

NP	DE					DE – MPFSC				
	F	CR	$\xi$	$\zeta$	$\epsilon\%$	F*	CR*	$\xi$	$\zeta$	$\epsilon\%$
400	0.1	0.9	0.77	0.43	$\pm 0.558$	0.1	0.9	0.15	0.015	$\pm 0.1$
400	/	/	0.85	0.29	$\pm 0.341$	0.4	0.5	0.25	0.02	$\pm 0.08$
400	/	/	0.81	0.55	$\pm 0.679$	0.2	0.2	0.20	0.05	$\pm 0.25$
400	/	/	0.80	0.63	$\pm 0.787$	0.2	0.3	0.30	0.042	$\pm 0.14$
400	/	/	0.79	0.51	$\pm 0.645$	0.2	0.3	0.22	0.02	$\pm 0.09$
Average	DE					DE – MPFSC				
	0.1	0.9	0.804	0.482	$\pm 0.602$	0.22	0.44	0.224	0.0294	$\pm 0.132$
Variance	DE					DE – MPFSC				
	0	0	0.00088	0.01672	$\pm 0.02802$	0.012	0.078	0.00313	0.00024	$\pm 0.00487$

**Table 3.** Numerical Analysis of Entanglement Degree  $\frac{1}{\xi}$  and Entanglement Degree Error  $\zeta$  of the DE algorithm and the DE-MPFSC algorithm 600 times.

NP	DE					DE – MPFSC				
	F	CR	$\xi$	$\zeta$	$\epsilon\%$	F*	CR*	$\xi$	$\zeta$	$\epsilon\%$
600	0.1	0.9	0.69	0.38	$\pm 0.550$	0.1	0.9	0.15	0.015	$\pm 0.1$
600	/	/	0.70	0.43	$\pm 0.614$	0.6	0.8	0.33	0.10	$\pm 0.303$
600	/	/	0.81	0.59	$\pm 0.728$	0.3	0.4	0.19	0.09	$\pm 0.474$
600	/	/	0.75	0.41	$\pm 0.547$	0.2	0.3	0.22	0.011	$\pm 0.05$
600	/	/	0.79	0.49	$\pm 0.620$	0.2	0.3	0.27	0.07	$\pm 0.259$
Average	DE					DE – MPFSC				
	0.1	0.9	0.748	0.46	$\pm 0.6118$	0.28	0.54	0.232	0.0572	$\pm 0.2372$
Variance	DE					DE – MPFSC				
	0	0	0.00282	0.0069	$\pm 0.0054$	0.037	0.083	0.00492	0.00175	$\pm 0.02869$

**Table 4.** Numerical Analysis of Entanglement Degree  $\frac{1}{\xi}$  and Entanglement Degree Error  $\zeta$  of the DE algorithm and the DE-MPFSC algorithm 800 times.

NP	DE					DE – MPFSC				
	F	CR	$\xi$	$\zeta$	$\epsilon\%$	F*	CR*	$\xi$	$\zeta$	$\epsilon\%$
800	0.1	0.9	0.63	0.44	$\pm 0.698$	0.1	0.9	0.15	0.015	$\pm 0.1$
800	/	/	0.64	0.31	$\pm 0.484$	0.5	0.4	0.27	0.03	$\pm 0.111$
800	/	/	0.75	0.50	$\pm 0.667$	0.3	0.3	0.33	0.06	$\pm 0.182$
800	/	/	0.70	0.50	$\pm 0.714$	0.2	0.3	0.30	0.017	$\pm 0.057$
800	/	/	0.71	0.36	$\pm 0.507$	0.2	0.3	0.27	0.02	$\pm 0.074$
Average	DE					DE – MPFSC				
	0.1	0.9	0.686	0.422	$\pm 0.614$	0.26	0.44	0.264	0.0284	$\pm 0.1048$
Variance	DE					DE – MPFSC				
	0	0	0.00253	0.00722	$\pm 0.01205$	0.023	0.068	0.00468	0.00035	$\pm 0.00231$

The above five data sheets in Tables 1–5 are data analysis of the entanglement degree  $\frac{1}{\xi}$ , entanglement degree error  $\zeta$  and error rate  $\epsilon$  about convergence precision and convergence speed. We experimented with 200 individuals as the cardinal number, repeating five times per experiment. We record the entanglement degree  $\frac{1}{\xi}$ , entanglement degree error  $\zeta$  and error rate  $\epsilon$  about DE algorithm and DE-MPFSC algorithm in real time. At the same time, we record the dynamic mutation factor F

and the dynamic crossover probability  $CR$  of the DE-MPFSC algorithm and analyze the data tables one by one.

**Table 5.** Numerical Analysis of Entanglement Degree  $\frac{1}{\xi}$  and Entanglement Degree Error  $\zeta$  of the DE algorithm and the DE-MPFSC algorithm 1000 times.

NP	DE					DE – MPFSC				
	$F$	$CR$	$\zeta$	$\zeta$	$\varepsilon\%$	$F^*$	$CR^*$	$\zeta$	$\zeta$	$\varepsilon\%$
1000	0.1	0.9	0.69	0.40	$\pm 0.579$	0.1	0.9	0.15	0.015	$\pm 0.1$
1000	/	/	0.82	0.66	$\pm 0.805$	0.3	0.4	0.30	0.04	$\pm 0.133$
1000	/	/	0.67	0.49	$\pm 0.731$	0.3	0.3	0.29	0.02	$\pm 0.068$
1000	/	/	0.77	0.58	$\pm 0.753$	0.2	0.3	0.27	0.019	$\pm 0.070$
1000	/	/	0.79	0.44	$\pm 0.557$	0.2	0.3	0.32	0.018	$\pm 0.056$
Average	DE					DE – MPFSC				
	0.1	0.9	0.748	0.514	$\pm 0.685$	0.22	0.44	0.266	0.0224	$\pm 0.0854$
Variance	DE					DE – MPFSC				
	0	0	0.00422	0.01118	$\pm 0.01219$	0.007	0.068	0.00453	0.0001	$\pm 0.00097$

In summary, when the number of individuals is 200, 400, 600, 800 or 1000, the range of entanglement error rate of the DE algorithm and the DE-MPFSC algorithm are respectively  $\varepsilon \in (\pm 0.714\%, \pm 0.904\%)$  and  $\varepsilon \in (\pm 0.133\%, \pm 0.578\%)$ , which shows that the entanglement degree of the DE-MPFSC algorithm is significantly higher than that of the DE algorithm, and the stability of the DE-MPFSC algorithm is better. The range of entanglement degree of the DE algorithm and the DE-MPFSC algorithm are respectively  $\frac{1}{\xi} \in (0.65, 0.9)$  and  $\frac{1}{\xi} \in (0.15, 0.35)$ , which shows that the entanglement degree of the DE-MPFSC algorithm is significantly higher than that of the DE algorithm, and the fluctuation of the DE-MPFSC algorithm is lower. The entanglement error ranges of the DE algorithm and the DE-MPFSC algorithm are respectively  $\zeta \in (0.25, 0.7)$  and  $\zeta \in (0.015, 0.1)$ , which shows that the entanglement degree of the DE-MPFSC algorithm is significantly higher than that of the DE algorithm, and the DE-MPFSC convergence of the algorithm is stronger. In addition, there IS an additional discovery during the experiment that the optimal dynamic mutation factor and the optimal dynamic crossover probability of the DE-MPFSC algorithm are  $F = 0.2$  and  $CR = 0.3$  respectively.

In summary, the convergence, stability and effectiveness of the DE-MPFSC algorithm are higher than the DE algorithm, which has obvious advantages for the integration of individuals.

#### 5.4. Test and Analysis of DE-MPFSC Algorithm about Several Classic DE Algorithms

##### 5.4.1. Entanglement Degree Test of DE-MPFSC Algorithm about Several Classic DE Algorithms

We choose four classic DE improved algorithms JDE, OBDE, DEGL and SADE for related tests. Through entanglement degree test and data analysis, we can verify the related performance of the DE-MPFSC algorithm and its superiority in data integration. From the previous experiments, we can know that the DE-MPFSC is better than the DE in stability, effectiveness and convergence. At the same time, the optimal dynamic mutation factor  $F = 0.2$  and the optimal crossover probability  $CR = 0.3$  of the DE-MPFSC algorithm were obtained through data experiments. In the following test experiments, we use them as the initial mutation factor and the initial cross probability to better compare with other classic DE algorithms. Because the function of data integration is closely related to the data size and dimension, the algorithm in this paper sets the number of individuals to 1000 and are divided into 5 experiments. The number of individuals gradually increases in each test and the test runs an average of 4 times. We record relevant experimental indicators: dynamic mutation probability  $F$ , dynamic cross probability  $CR$ , entanglement degree  $\frac{1}{\xi}$ , entanglement error  $\zeta$ , error rate  $\varepsilon\%$ , average individual loss efficiency. Figures 11–15 below shows a numerical results of the experiment.

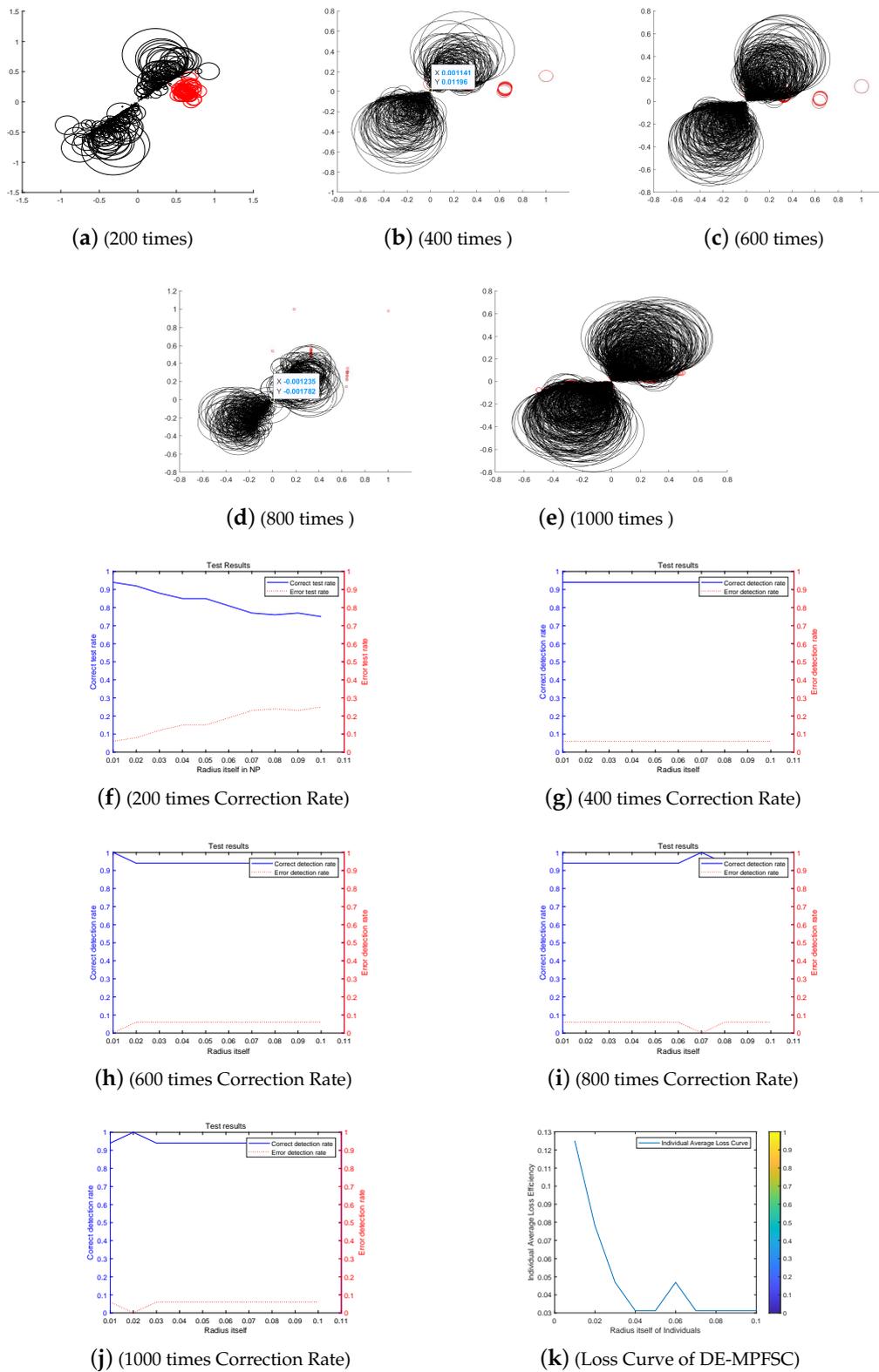


Figure 11. The Entanglement Degree, Correction Rate and Loss Curve of DE-MPFSC algorithm.

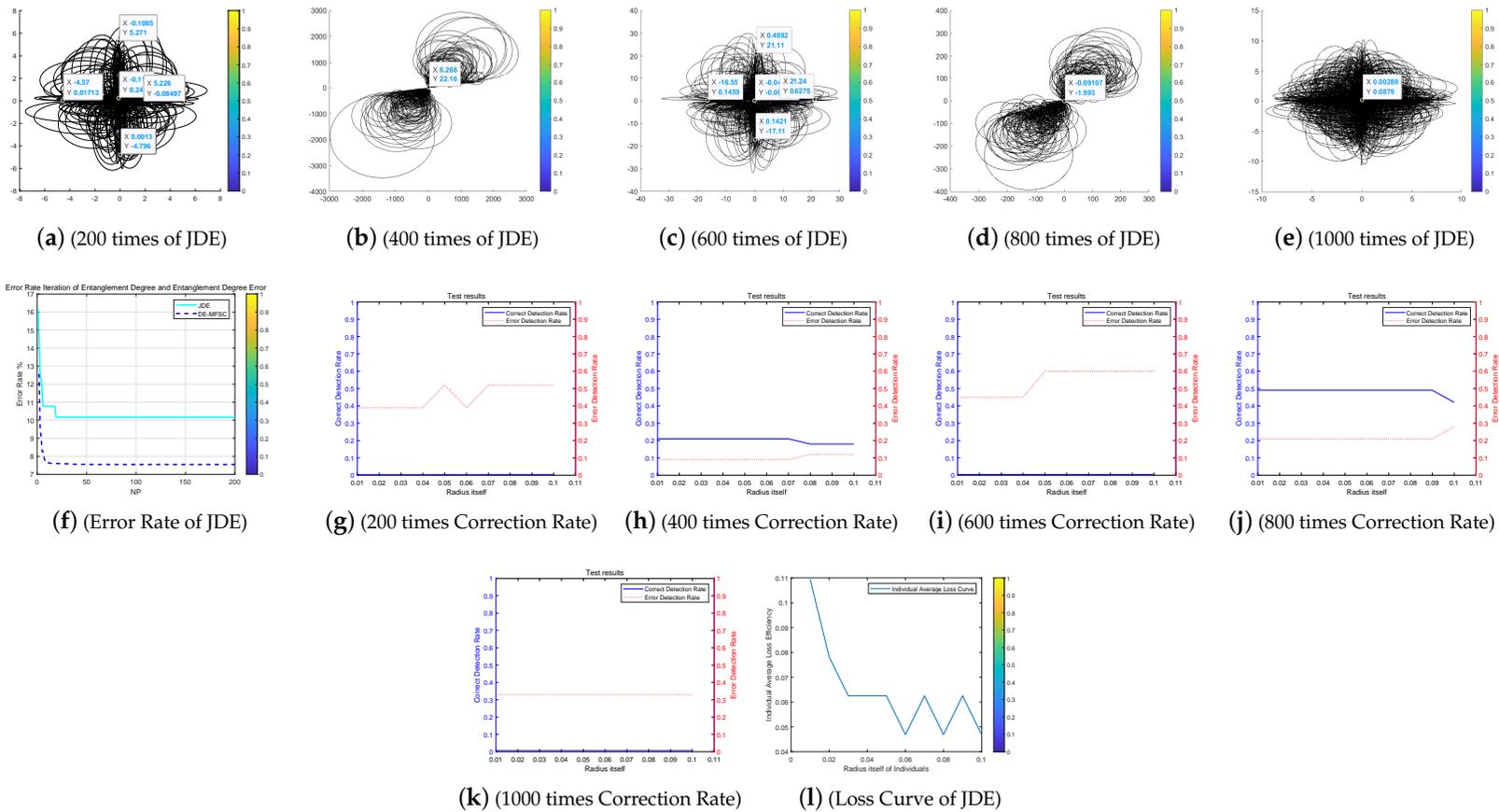


Figure 12. The Entanglement Degree, Correction Rate and Loss Curve of the (Self-Adapting Parameter Setting in Differential Evolution) JDE algorithm.

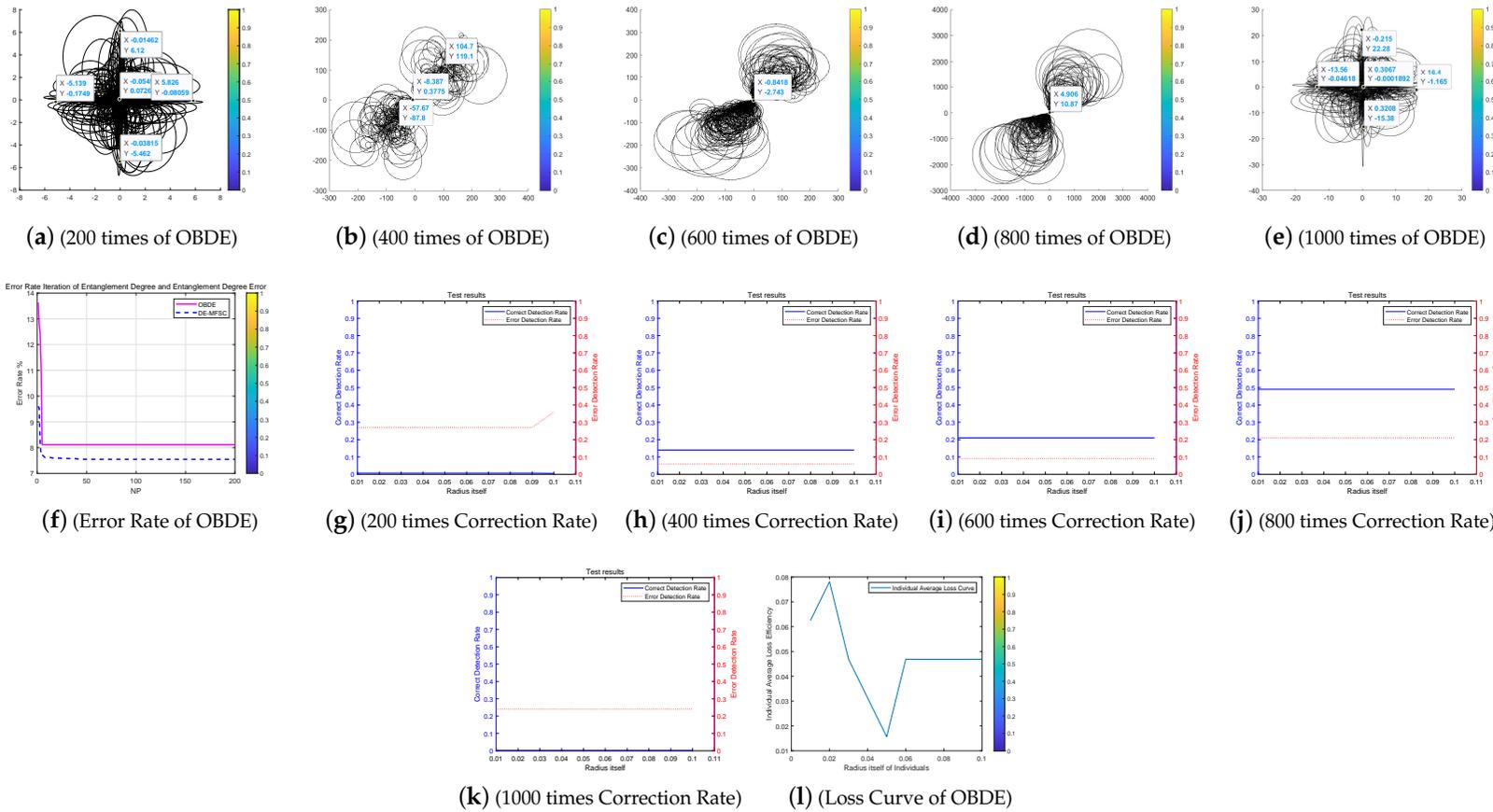


Figure 13. The Entanglement Degree, Correction Rate and Loss Curve of the Opposition Based Differential Evolution (OBDE) algorithm.

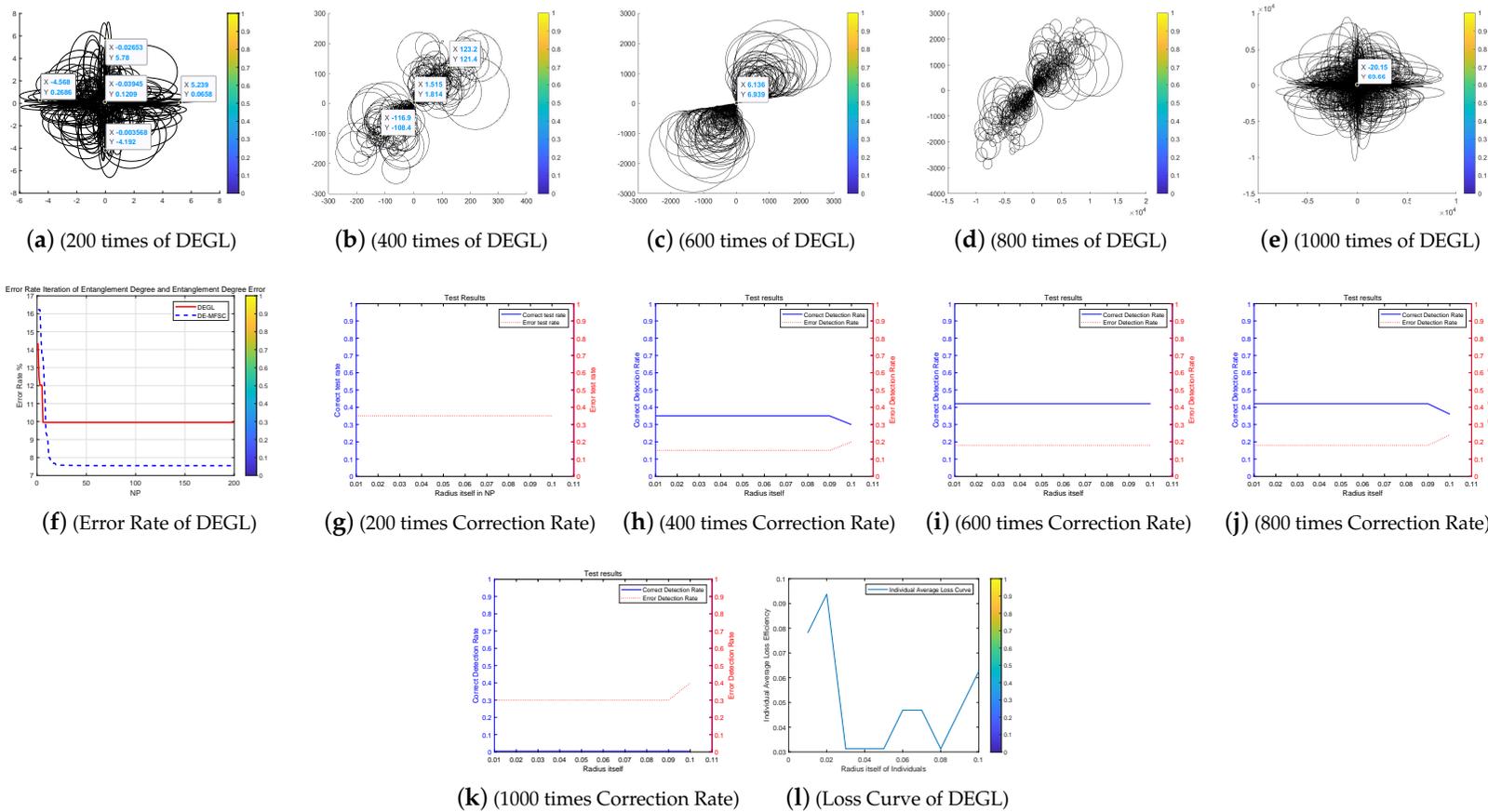


Figure 14. The Entanglement Degree, Correction Rate and Loss Curve of the (Differential Evolution with Global and Local neighborhoods) DEGL algorithm.

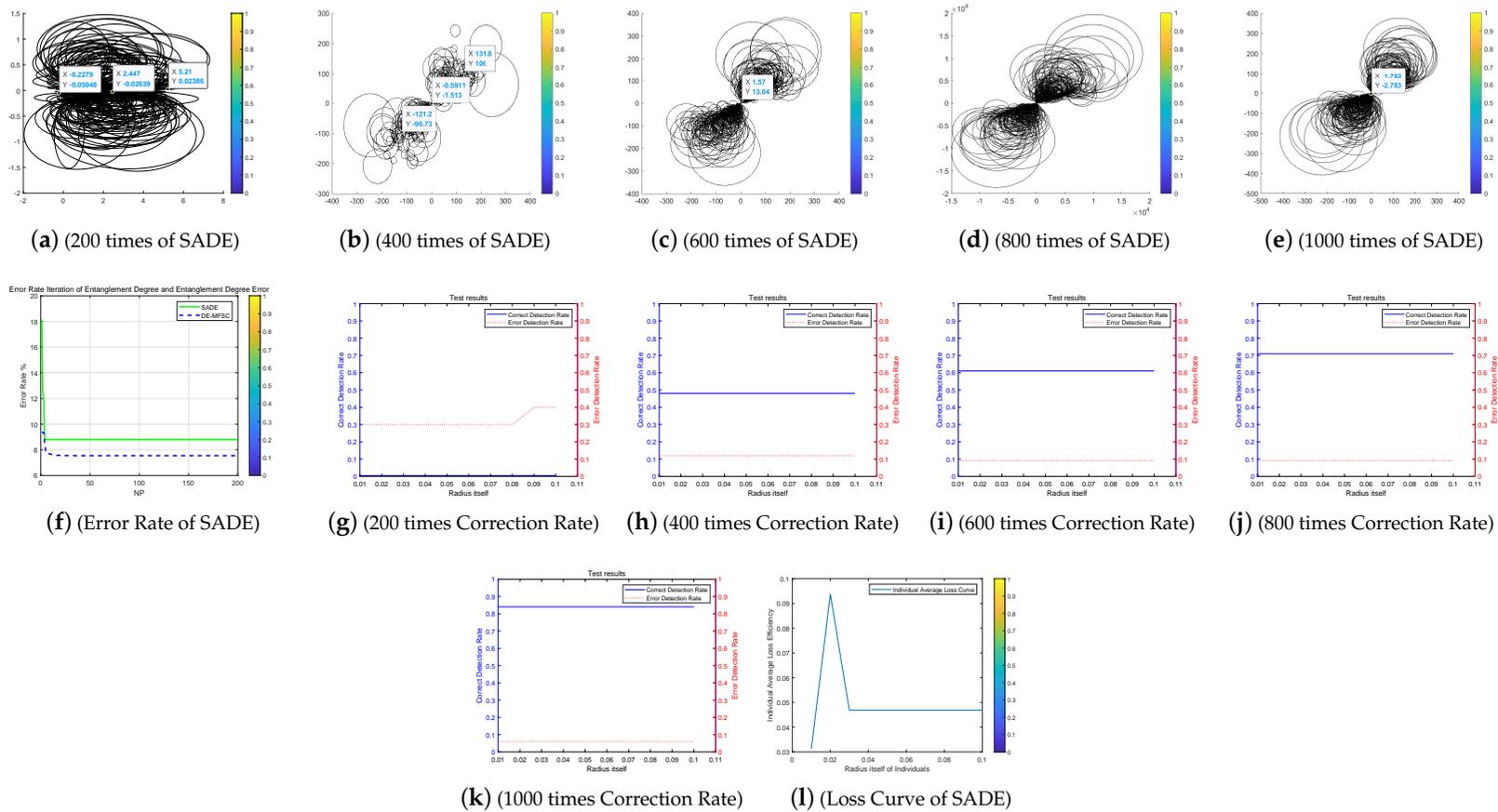


Figure 15. The Entanglement Degree, Correction Rate and Loss Curve of the (Self-Adaptive Differential Evolution) SADE algorithm.

We analyze the above Figures 11–15. First we analyze from the entanglement degree. In the process of the DE-MPFSC algorithm, the convergence speed and convergence precision of population individuals gradually stabilized as the population increased. The density and integration of the two entangled images gradually increase and, at the same time, the edge differentiation phenomenon is further weakened on each branch, which make the entangled branch converge to the center of the entanglement along the convergence curve. Compared with the DE-MPFSC algorithm, the entanglement phenomenon of the convergence speed and convergence precision of the JDE algorithm is less obvious. When the number of individuals is 200, 600 and 1000, the entangled image is highly dispersed, which causes a severe edge differentiation on the entangled branches. Although, when the number of individuals is 400 and 600, the state of entanglement is better than that of 200, 600 and 800, the continuity of individual evolution is not reached as a whole, it does not converge to the center of entanglement along the convergence curve. JDE's entangled image shows irregular fluctuations. The OBDE algorithm is slightly better than the JDE algorithm in fluctuation, when the number of individuals is 200 and 1000, the entangled branches are severely differentiated and the dispersed degree is higher than that of JDE, which shows that the OBDE algorithm is not suitable. When the number of individuals is 400, 600 and 800, the dispersed degree of the entangled branches of the OBDE algorithm is lower but the edge dispersion of the entangled branches is more obvious than the DE-MPFSC algorithm. OBDE and JDE algorithms also do not show evolutionary continuity. The performance of the DEGL algorithm on evolutionary fluctuation is weaker than JDE and OBDE. When the number of individuals is 200 and 1000, the entangled branches are highly dispersed and do not converge to the entangled center. The DEGL is similar to the JDE algorithm and the OBDE algorithm. The individual convergence speed and convergence accuracy completely deviate from the entanglement center. The number of entangled branches integrated with 400 and 800 individuals is much higher than that with 200 and 1000 individuals. Although DEGL converges to the center of entanglement along the convergence curve, the edge differentiation of entangled branches is still not resolved. The continuity, volatility and convergence of the SADE algorithm are the closest to the DE-MPFSC algorithm but, unfortunately, a discrete state occurs when the number of individuals is 200, which makes the entanglement center distributed on the fitted line rather than an entanglement center. When the number of individuals is 400, 600, 800 and 1000, the edge dispersion of the entangled branches is weakened but it is not used for large-scale evolution.

Then we analyze from the correct test rate of entangled branches. The entanglement degree of the DE-MPFSC algorithm is divided into two branches about convergence precision and convergence speed. From the image, the convergence speed and convergence precision are lower discrete and they converge to the entanglement center point along the convergence curve. The detection efficiency are 3, 19, 19, 19, 19, which shows that when the population individuals are 200, 400, 600, 800, 1000, the stability of the DE-MPFSC algorithm is higher. Due to the fluctuation of the JDE algorithm, its correct test rate is 0, and its error test rates are 0.4, 0.4, 0.95, 0.6 and 0.95, respectively. JDE's higher error test rate is easy to eliminate optimal individuals. The OBDE algorithm is weaker than the JDE algorithm in fluctuation. Its correct test rate is 0 but its error test rates are 1, 1, 0.4, 0.4, 1, which are higher than that of the JDE algorithm. When the number is 1000, the error test rate suddenly rebounds. The DEGL algorithm has a non-zero in correct test rate, which at least shows that there is a certain ability to adapt to individual mutations. But when the number of individuals is 200 and 1000, the error test rate of entangled branches is 1 and 0.95, which is still too high. When the number of individuals is 400 and 800, the correct test rate of entangled branches declines. The instability of this entanglement behavior is still not conducive to the evolution of mutation individuals. The correct test rate of the SADE algorithm are 0, 0, 0.35, 0.38, 0.35, and the error test rates are 0.4, 0.15, 0.15, 0.2, 0.15, which are much better than the JDE, OBDE and DEGL algorithms. Stability and continuity should be more conducive to the evolutionary behavior of large-scale individuals. However, compared to the DE-MPFSC algorithm, because of the correct test rate of the DEGL algorithm being too low, the detection of mutation individuals is not adaptive.

Finally, from the individual loss curve, we can know that the fluctuation of the JDE is higher than that of DE-MPFSC, OBDE, DEGL and SADE, indicating that during the individual evolution process, the population individual loss degree is higher. Especially for large-scale evolutionary populations, the risk of mutation individuals being lost during evolution will greatly increase. The individual loss degree of OBDE and DEGL rebounded in the later stages of evolution. The loss degree of mutation individuals showed an increasing trend relative to DE-MPFSC, JDE and SADE. The individual loss of SADE and DE-MPFSC is similar but the individual loss of SADE is higher than the latter. Because the individual losses of JDE, OBDE, DEGL and SADE are higher than DE-MPFSC, the advantages of the latter are obvious in stability. By analyzing the error rate curve, we can know that when DE-MPFSC is processing large-scale individuals (based on 200 individuals), the error rate of individual loss is lower than other algorithms.

#### 5.4.2. Numerical Test of DE-MPFSC Algorithm about Several Classic DE Algorithms

We analyze the test results of the above data Table 6. We already know that when the number of individuals is 200, 400, 600, 800, 1000, the entanglement error range of DE-MPFSC is  $\varepsilon \in (\pm 0.133\%, \pm 0.578\%)$ , The entanglement range of DE-MPFSC is  $\frac{1}{\xi} \in (0.15, 0.35)$ , and the entanglement error range of DE-MPFSC is  $\zeta \in (0.015, 0.1)$ . When the number of individuals is 200, the entanglement error rate ranges of JDE, OBDE, DEGL and SADE are  $\varepsilon \in (\pm 0.4\%, \pm 0.45\%)$ ,  $\varepsilon = \pm 1\%$ ,  $\varepsilon = \pm 1\%$ ,  $\varepsilon \in (\pm 0.3\%, \pm 0.4\%)$ , compared with DE-MPFSC, the former has a higher entanglement error rate, which shows that the entanglement degree of DE-MPFSC (since we have done a detailed analysis of the entanglement effect of the DE-MPFSC algorithm in the case of the same number of individuals, we will not repeat it here. For all comparative data, see Section 5.3) is significantly higher than that of JDE, OBDE, DEGL and SADE, and the algorithm has good stability and effectiveness. We also found a special case, when the number of individuals is 200, the entanglement degree and entanglement degree error of JDE, OBDE, DEGL are infinite or non-existent. The reason for the case is that the entangled branches are too discrete or the edges of the entangled branches are too severely differentiated. Obviously, their entanglement efficiency is significantly lower than DE-MPFSC. The entanglement range and entanglement error of SADE are  $\frac{1}{\xi} \in (2.43 \times 10^{-2}, 1.06 \times 10^{-1})$  and  $\zeta \in (2.43 \times 10^{-2}, 1.06 \times 10^{-1})$ , which is also lower than the entanglement effect of DE-MPFSC, which shows that the degree of entanglement of DE-MPFSC is significantly higher than that of JDE, OBDE, DEGL and SADE. The algorithm has small fluctuations and good convergence. When the number of individuals is 400, the entanglement error rate ranges of JDE, OBDE, DEGL and SADE are  $\varepsilon \in (\pm 0.4\%, \pm 0.45\%)$ ,  $\varepsilon = \pm 1\%$ ,  $\varepsilon = \pm 0.25\%$ ,  $\varepsilon = \pm 0.15\%$ , compared with DE-MPFSC, the entanglement error rate is higher, which shows that the degree of entanglement of DE-MPFSC is significantly higher than that of JDE, OBDE, DEGL and SADE, and the algorithm is better stability and effectiveness. The entanglement ranges of JDE, OBDE, DEGL and SADE are  $\frac{1}{\xi} \in (5.44 \times 10^{-2}, 2.33 \times 10^{-1})$ ,  $\frac{1}{\xi} \in (0.53 \times 10^0, 4.22 \times 10^{-2})$ ,  $\frac{1}{\xi} \in (1.73 \times 10^{-1}, 2.04 \times 10^{-1})$ ,  $\frac{1}{\xi} \in (6.23 \times 10^{-2}, 2.43 \times 10^{-1})$ , which shows that the degree of entanglement of the DE-MPFSC is significantly higher than that of JDE, OBDE, DEGL, SADE, and the DE-MPFSC algorithm has less volatility. The entanglement degree error ranges of JDE, OBDE, DEGL and SADE are  $\zeta \in (1.13 \times 10^{-3}, 1.49 \times 10^{-1})$ ,  $\zeta \in (0.59 \times 10^0, 0.69 \times 10^{-1})$ ,  $\zeta \in (1.34 \times 10^{-1}, 3.70 \times 10^{-2})$ ,  $\eta \in (2.95 \times 10^{-2}, 1.08 \times 10^{-1})$ , which shows that the degree of entanglement of DE-MPFSC is significantly higher than that of JDE, OBDE, DEGL and SADE, and the algorithm has strong convergence. When the number of individuals is 600, 800, 1000, the comparison between DE-MPFSC and JDE, OBDE, DEGL and SADE is similar, and the data has been marked in the Table 6. On the whole, the effectiveness, stability, and convergence of DE-MPFSC are higher than JDE, OBDE, DEGL and SADE on average.

**Table 6.** Numerical Analysis of Entanglement Degree  $\frac{1}{\zeta}$  and Entanglement Degree Error  $\zeta$  of the DE algorithm and the DE-MPFSC algorithm Several Times.

NP	JDE					OBDE					DEGL					SADE				
	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$
200	0.2	0.3	$\infty$	$\infty$	$\pm 0.40$	0.2	0.3	–	–	$\pm 1$	0.2	0.3	$\infty$	$\infty$	$\pm 1$	0.2	0.3	$1.04 \times 10^{-1}$	$1.04 \times 10^{-1}$	$\pm 0.30$
200	/	/	$\infty$	$\infty$	$\pm 0.45$	/	/	–	–	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$1.06 \times 10^{-1}$	$1.06 \times 10^{-1}$	$\pm 0.40$
200	/	/	$\infty$	$\infty$	$\pm 0.40$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	–	–	$\pm 1$	/	/	$0.81 \times 10^{-1}$	$0.81 \times 10^{-1}$	$\pm 0.35$
200	/	/	–	–	$\pm 0.40$	/	/	–	–	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$2.43 \times 10^{-2}$	$2.43 \times 10^{-2}$	$\pm 0.35$
200	/	/	–	–	$\pm 0.45$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	–	$\infty$	$\pm 1$	/	/	$2.77 \times 10^{-2}$	$2.77 \times 10^{-2}$	$\pm 0.40$
NP	JDE					OBDE					DEGL					SADE				
	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$
400	0.2	0.3	$0.88 \times 10^0$	$0.49 \times 10^0$	$\pm 0.45$	0.2	0.3	$0.53 \times 10^0$	$0.60 \times 10^0$	$\pm 1$	0.2	0.3	$2.04 \times 10^{-1}$	$1.34 \times 10^{-1}$	$\pm 0.25$	0.2	0.3	$2.42 \times 10^{-1}$	$1.06 \times 10^{-1}$	$\pm 0.15$
400	/	/	$5.44 \times 10^{-2}$	$1.49 \times 10^{-1}$	$\pm 0.40$	/	/	$0.58 \times 10^0$	$0.59 \times 10^0$	$\pm 1$	/	/	$1.74 \times 10^{-1}$	$1.41 \times 10^{-1}$	$\pm 0.25$	/	/	$2.41 \times 10^{-1}$	$1.05 \times 10^{-1}$	$\pm 0.15$
400	/	/	$6.05 \times 10^{-2}$	$0.52 \times 10^0$	$\pm 0.45$	/	/	$1.61 \times 10^{-1}$	$4.72 \times 10^{-2}$	$\pm 1$	/	/	$1.72 \times 10^{-1}$	$3.72 \times 10^{-2}$	$\pm 0.25$	/	/	$6.23 \times 10^{-2}$	$2.95 \times 10^{-2}$	$\pm 0.15$
400	/	/	$2.15 \times 10^{-1}$	$0.47 \times 10^0$	$\pm 0.40$	/	/	$4.22 \times 10^{-2}$	$4.51 \times 10^{-2}$	$\pm 1$	/	/	$1.73 \times 10^{-1}$	$1.38 \times 10^{-1}$	$\pm 0.25$	/	/	$6.49 \times 10^{-2}$	$3.02 \times 10^{-1}$	$\pm 0.15$
400	/	/	$2.33 \times 10^{-1}$	$1.13 \times 10^{-3}$	$\pm 0.40$	/	/	$1.67 \times 10^{-1}$	$1.69 \times 10^{-1}$	$\pm 1$	/	/	$5.16 \times 10^{-2}$	$3.70 \times 10^{-2}$	$\pm 0.25$	/	/	$2.43 \times 10^{-1}$	$1.08 \times 10^{-1}$	$\pm 0.15$
NP	JDE					OBDE					DEGL					SADE				
	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$
600	0.2	0.3	–	–	$\pm 1$	0.2	0.3	$4.44 \times 10^{-2}$	$0.49 \times 10^0$	$\pm 0.45$	0.2	0.3	$1.62 \times 10^{-1}$	$3.71 \times 10^{-2}$	$\pm 0.15$	0.2	0.3	$2.20 \times 10^{-1}$	$1.11 \times 10^{-1}$	$\pm 0.1$
600	/	/	–	–	$\pm 1$	/	/	$1.62 \times 10^{-1}$	$0.51 \times 10^0$	$\pm 0.46$	/	/	$4.30 \times 10^{-2}$	$1.38 \times 10^{-1}$	$\pm 0.1$	/	/	$2.18 \times 10^{-1}$	$1.10 \times 10^{-1}$	$\pm 0.1$
600	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$1.62 \times 10^{-1}$	$0.47 \times 10^0$	$\pm 0.40$	/	/	$1.56 \times 10^{-1}$	$3.70 \times 10^{-2}$	$\pm 0.1$	/	/	$2.17 \times 10^{-1}$	$2.67 \times 10^{-2}$	$\pm 0.15$
600	/	/	–	–	$\pm 1$	/	/	$4.41 \times 10^{-2}$	$1.38 \times 10^{-1}$	$\pm 0.40$	/	/	$4.28 \times 10^{-2}$	$1.36 \times 10^{-1}$	$\pm 0.15$	/	/	$5.91 \times 10^{-2}$	$1.02 \times 10^{-1}$	$\pm 0.15$
600	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$4.49 \times 10^{-2}$	$1.16 \times 10^{-1}$	$\pm 0.38$	/	/	$1.59 \times 10^{-1}$	$3.69 \times 10^{-2}$	$\pm 0.1$	/	/	$2.17 \times 10^{-1}$	$1.00 \times 10^{-1}$	$\pm 0.1$
NP	JDE					OBDE					DEGL					SADE				
	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$
800	0.2	0.3	$5.49 \times 10^{-2}$	$1.41 \times 10^{-1}$	$\pm 0.6$	0.2	0.3	$2.28 \times 10^{-1}$	$0.30 \times 10^0$	$\pm 0.40$	0.2	0.3	$2.01 \times 10^{-2}$	$1.31 \times 10^{-1}$	$\pm 0.30$	0.2	0.3	$2.14 \times 10^{-1}$	$1.01 \times 10^{-1}$	$\pm 0.15$
800	/	/	$5.12 \times 10^{-2}$	$1.44 \times 10^{-1}$	$\pm 0.6$	/	/	$2.25 \times 10^{-1}$	$0.35 \times 10^0$	$\pm 0.30$	/	/	$2.03 \times 10^{-1}$	$1.38 \times 10^{-1}$	$\pm 0.40$	/	/	$2.15 \times 10^{-1}$	$1.06 \times 10^{-1}$	$\pm 0.15$
800	/	/	$15.70 \times 10^{-3}$	$3.84 \times 10^{-2}$	$\pm 0.6$	/	/	$2.36 \times 10^{-1}$	$2.54 \times 10^{-2}$	$\pm 0.40$	/	/	$2.03 \times 10^{-1}$	$3.72 \times 10^{-2}$	$\pm 0.50$	/	/	$2.08 \times 10^{-1}$	$2.82 \times 10^{-2}$	$\pm 0.2$
800	/	/	$2.06 \times 10^{-1}$	$3.99 \times 10^{-2}$	$\pm 0.6$	/	/	$2.17 \times 10^{-1}$	$0.37 \times 10^0$	$\pm 0.30$	/	/	$2.05 \times 10^{-1}$	$1.34 \times 10^{-1}$	$\pm 0.60$	/	/	$5.57 \times 10^{-2}$	$0.95 \times 10^{-1}$	$\pm 0.2$
800	/	/	$15.86 \times 10^{-3}$	$1.54 \times 10^{-1}$	$\pm 0.6$	/	/	$2.22 \times 10^{-1}$	$1.01 \times 10^{-1}$	$\pm 0.40$	/	/	$5.55 \times 10^{-2}$	$3.62 \times 10^{-2}$	$\pm 0.75$	/	/	$1.52 \times 10^{-1}$	$0.96 \times 10^{-1}$	$\pm 0.2$
NP	JDE					OBDE					DEGL					SADE				
	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$	F	CR	$\zeta$	$\zeta$	$\epsilon\%$
1000	0.2	0.3	–	–	$\pm 1$	0.2	0.3	$\infty$	$\infty$	$\pm 1$	0.2	0.3	$\infty$	$\infty$	$\pm 1$	0.2	0.3	$1.93 \times 10^{-1}$	$1.03 \times 10^{-1}$	$\pm 0.1$
1000	/	/	–	–	$\pm 0.1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$1.88 \times 10^{-1}$	$0.96 \times 10^{-1}$	$\pm 0.1$
1000	/	/	$\infty$	$\infty$	$\pm 1$	/	/	–	–	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$1.90 \times 10^{-1}$	$2.60 \times 10^{-2}$	$\pm 0.1$
1000	/	/	$\infty$	$\infty$	$\pm 1$	/	/	–	–	$\pm 1$	/	/	–	–	$\pm 1$	/	/	$5.09 \times 10^{-2}$	$0.87 \times 10^{-1}$	$\pm 0.1$
1000	/	/	–	–	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$\infty$	$\infty$	$\pm 1$	/	/	$1.91 \times 10^{-1}$	$0.84 \times 10^{-1}$	$\pm 0.1$

Then, we analyze from the state of entangled branches. When the number of individuals is 200, 400 or 1000, the entanglement degree and entanglement degree error of JDE do not exist, and the entanglement branches are in a discrete state. As the number of individuals increases, the entanglement effect appears fluctuating. When the number of individuals is 200 or 1000, the entanglement degree and entanglement degree error of OBDE do not exist, and the entangled branches are in a discrete state. When the number of individuals is 200, 400, 600 or 800, the integration of entangled branches gradually increases. However, the increase of OBDE is weaker than that of DE-MPFSC. The entanglement of the former shows a coexistence state of gradualism and fluctuation. When the number of individuals is 200 or 1000, the entanglement degree and entanglement degree error of DEGL do not exist, and the entangled branches are in a discrete state. When the number of individuals is 400 and 800, the integration of entangled branches begins to appear and it reaches the best state when the number of individuals is 600. The entanglement of DEGL shows coexistence of symmetry and fluctuation. The case is not conducive to the stable development of population evolution. When the number of individuals gradually increases, the entanglement degree and entanglement degree error of SADE are always integrated and increase with the number of individuals. However, due to the dispersion of the edges of the entangled branches, the entanglement increases slowly. Overall, DE-MPFSC has better stability and effectiveness than JDE, OBDE, DEGL and SADE.

## 6. Empirical Analysis

### 6.1. Verification Data Sets

We consider the imbalanced datasets, the UCI machine learning datasets [51] (the UCI Machine Learning Database is available at the following URL. <https://archive.ics.uci.edu/ml/index.php>), as integrated datasets to test the practical application level of the DE-MPFSC algorithm. First, we select three types of testing data from the UCI machine learning datasets, which are numerical value datasets (N.V. datasets), classify value datasets (C.V. datasets) and mixed value datasets (M.V. datasets). At the same time, according to the data integration level, the above three types of datasets are divided into level I, level II and level III, their proportions are 1, 2 and 3, respectively. The specific information on the three types of datasets is shown in Table 7.

**Table 7.** The situation description of various types of data about the UCI machine learning datasets.

Datasets	Sample Capacity	N.V. Datasets	C.V. Datasets	M.V. Datasets
Dermatology	400	5	30	8
Credit Approval	572	9	13	4
Automobile	382	15	27	17
Sponge	108	6	32	28
Contraceptive Method Choice (CMC)	1458	4	21	16
Soybean	453	2	39	14
Glass	314	5	4	12

In this experiment, we divided the three types of data integration into single-link data integration (SLCE) and complete-link data integration (CLCE) according to the DE-MPFSC algorithm. The results obtained by the two integrated methods are compared with the results obtained by the KNN-SK (KNN data filling) [52–54] and the SKNN-SK (SKNN data filling) [52–54]. To ensure that there is single-random for the data in Table 1, we performed random deletion in a five percent ratio on the computer. Then, we conducted the algorithm test and obtained the results by performing the test 200 times on the computer in the same integrated method for the same datasets.

### 6.2. Verification Indicator

To interpret the effect of the DE-MPFSC algorithm on the imbalanced data integrated and classified, we need to verify the consistency of the imbalanced data. First, we should establish the verification indicator of classification accuracy (CA) [55,56], adjusted rand index (ARI) [55,56], normalized mutual information (NMI) [55,56] to test the effect of the DE-MPFSC algorithm, which are described as follows.

(1) CA: CA is a statistical mathematics variable measuring the imbalanced data samples proportion of the DE-MPFSC algorithm in making up for data integrated defects, which is expressed as follows.

$$CA = \frac{\sum_{i=1}^k \alpha_i}{n}, \tag{34}$$

where  $\alpha_i$  is a subsample of the DE-MPFSC algorithm in making up for data integrated defects,  $k$  is the number of integrated classes,  $n$  is the sample capacity.

(2) ARI: ARI is a statistical mathematics variable measuring the imbalanced data samples proportion of the DE-MPFSC algorithm in making up for data integrated defects after considering the same data class and different data classes, which is expressed as follows.

$$ARI = \frac{\sum_{i=1}^I \sum_{j=1}^J C_{n_{ij}}^2 - \eta}{\frac{1}{2}(\rho + \phi) - \eta} \tag{35}$$

$$\rho = \sum_{i=1}^I C_{n_i}^2, \phi = \sum_{j=1}^J C_{n_j}^2, \eta = \frac{2\rho\phi}{n(n-1)}. \tag{36}$$

(3) NMI: NMI is a similarity variable measuring sample data and integrated data after comparing and correctly integrating and classing imbalanced data within being integrated defects, which is expressed as follows.

$$NMI = \frac{\sum_{i=1}^I \sum_{j=1}^J \frac{n_{ij}^2 n}{n_i n_j}}{\sqrt{\sum_{i=1}^I \frac{n_i^2}{n} \sum_{j=1}^J \frac{n_j^2}{n}}}, \tag{37}$$

where  $n_{ij}$  is the sample size in the data integrated result where the  $i$ th datasets contains the original datasets  $j$ .  $n_i$  is the sample size of the  $i$ th datasets in the data integrated result.  $n_j$  is the sample size of the original datasets  $j$ .  $n$  is the samples.  $I, J$  is the number of different types of datasets and the number of original datasets obtained in the data integrated result. The limit value of the three types of verification indicators is set to 1. If the data structure is close to the original data structure after imbalanced integrated data, which means that the value of the dataset is larger compared to the original datasets. That is, the DE-MPFSC algorithm has a better effect on imbalanced data integration.

### 6.3. The Test Analysis

The number of neighbors selected by the KNN-SK and SKNN-SK is  $K = 10$ . The testing results of the (Differential Evolution Algorithm Based on Mixed Penalty Function Screening Criteria)DE-MPFSC algorithm and other algorithms under the verification indicators are shown in Table 8 and Figures 16–18 about Classification Accuracy (CA), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI) datasets.

**Table 8.** Data integration Comparison about CA, NMI and ARI Values of the DE-MPFSC Algorithm.

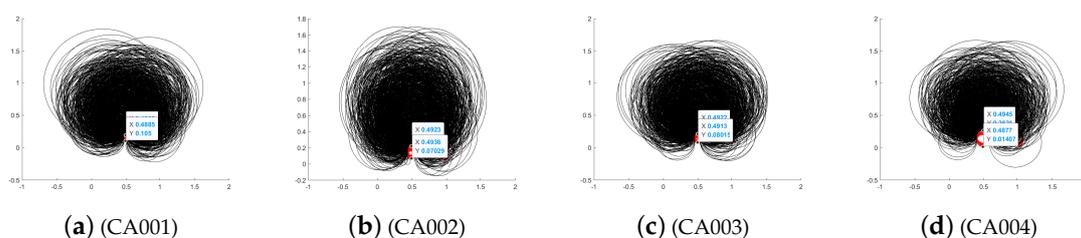
Data Sets	CA				ARI				NMI			
	SLCE	CLCE	KNN	SKNN	SLCE	CLCE	KNN	SKNN	SLCE	CLCE	KNN	SKNN
Der.	0.430	0.772	0.674	0.710	0.152	0.667	0.666	0.458	0.351	0.728	0.588	0.579
C.A.	0.576	0.753	0.726	0.756	0.261	0.275	0.215	0.274	0.206	0.217	0.167	0.213
Aut.	0.523	0.537	0.529	0.516	0.168	0.148	0.157	0.153	0.265	0.257	0.237	0.269
Spo.	0.780	0.790	0.738	0.726	0.424	0.438	0.523	0.447	0.705	0.701	0.707	0.747
■(CMC)	0.428	0.429	0.427	0.410	0.016	0.501	0.516	0.424	0.032	0.039	0.020	0.015
Soy.	0.545	0.668	0.631	0.634	0.431	0.436	0.356	0.346	0.622	0.637	0.638	0.696
■Glass	0.473	0.546	0.524	0.525	0.199	0.154	0.158	0.168	0.323	0.302	0.315	0.226
Average	0.322	0.262	0.267	0.183	0.783	0.203	0.358	0.283	0.437	0.392	0.438	0.395
Variance	0.002	0.0012	0.120	0.050	0.030	0.070	0.090	0.030	0.0013	0.0042	0.0912	0.095

Where according to the data analysis in Table 8, we know that since the single data integrated method KNN-SK and SKNN-SK do not have fitting characteristics inside the data integration and consistency are better than single integrated methods.

The CA value, ARI value and NMI value of UCI data about DE-MPFSC algorithm are as follows: the optimal value and sub-optimal value tend to distribution in probability 1, while the other data points tend to distribution in probability 0, in which the data points tended to distribution in probability 1 are in the feasible region and the data points tended to distribution in probability 0 are in the non-feasible region. The CA value, ARI value and NMI value of Credit Approval data about DE-MPFSC algorithm are as follows: the optimal value and sub-optimal value tend to distribution in probability 1, while the other data points tend to distribution in probability 0, in which the data points tended to distribution in probability 1 are in the feasible region and the data points tended to distribution in probability 0 are in the non-feasible region. For the data (CMC) and Glass marked with ■, the values of CA, ARI and NMI are as follows: the optimal value and the sub-optimal value are stable around the probability of 0.5, and the error does not exceed  $\pm 0.2$ , which accord with the stability requirements of data.

The CA value, ARI value and NMI value of all data about DE-MPFSC algorithm are as follows: the average optimal value and the average sub-optimal quality are stable around the probability of 0.5 and the error does not exceed  $\pm 0.2$ . The variance optimal value and the variance sub-optimal value are basically stable around 0.01, which accord with the data stability standard and the error does not exceed 0.0001. From the datasets in Table 8, the optimal value of the DE-MPFSC algorithm in both mean and variance shows the superiority of the algorithm.

It can be seen from the Figures 16–18 that through the two-dimensional reduced order analysis of UCI machine learning data, it is found that the three data integration effects of CA, NMI and ARI show a trend of highly integrated convergence within an acceptable error range.



**Figure 16.** Convergence integration of DE-MPFSC algorithm for CA.

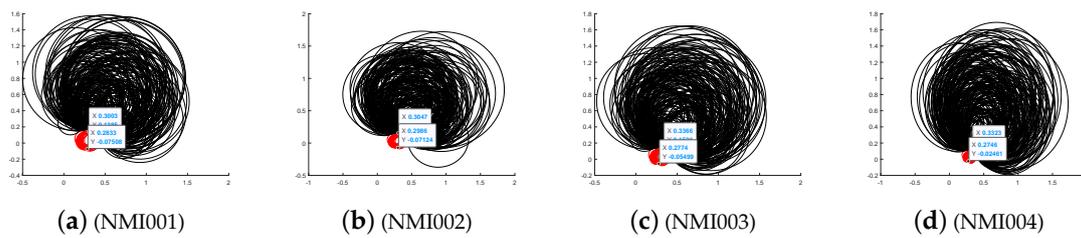


Figure 17. Convergence integration of DE-MPFFC algorithm for NMI.

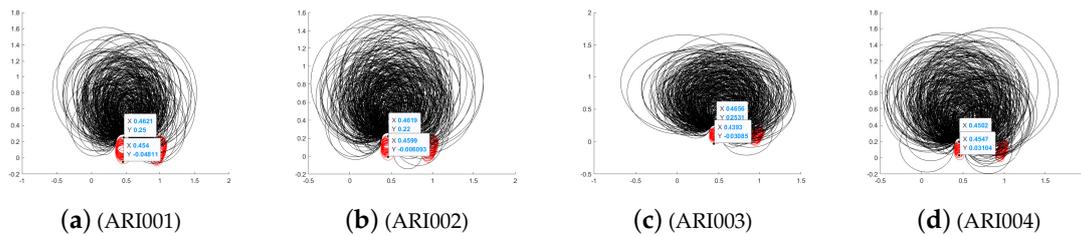


Figure 18. Convergence integration of DE-MPFFC algorithm for ARI.

First, we analyze from the precision. The integration effect on CA data is as follows: the minimum upper bound of the reduced-order 2D data in the X direction is  $\Omega_{min} = 0.4$  and the error does not exceed  $\pm 0.01$ . The maximum lower bound of the reduced-order 2D data in the Y direction is  $\Omega_{max} = 0.1$  and the error does not exceed  $\pm 0.02$ . The entanglement error of the imbalanced data set in both X and Y directions is  $\frac{\Omega_{max}}{\Omega_{min}} = 0.25$ . It can be seen that the precision in both directions of X and Y is in a fully controllable and highly integrated range, which fully shows that the DE-MPFFC algorithm has a good integration advantage for improving the integration accuracy of CA data. The integration effect on NMI data is as follows—the minimum upper bound of the reduced-order 2D data in the X direction is  $\Omega_{min} = 0.3$  and the error does not exceed  $\pm 0.03$ . The maximum lower bound of the reduced-order 2D data in the Y direction is  $\Omega_{max} = 0.02$  and the error does not exceed  $\pm 0.01$ . The entanglement error of the imbalanced data set in both X and Y directions is  $\frac{\Omega_{max}}{\Omega_{min}} = 0.06$ . It can be seen that the precision in both directions of X and Y is in a fully controllable and highly integrated range, which fully demonstrates that the DE-MPFFC algorithm has a good integration advantage for improving the integration accuracy of NMI data. The integration effect on ARI data is as follows: the minimum upper bound of the reduced-order 2D data in the X direction is  $\Omega_{min} = 0.45$  and the error does not exceed  $\pm 0.01$ . The maximum lower bound of the reduced-order 2D data in the Y direction is  $\Omega_{max} = 0.006$  and the error does not exceed  $\pm 0.01$ . The entanglement error of the imbalanced data set in both X and Y directions is  $\frac{\Omega_{max}}{\Omega_{min}} = 0.013$ . It can be seen that the precision in both directions of X and Y is in a fully controllable and highly integrated range, which fully shows that the DE-MPFFC algorithm has a good integration advantage for improving the integration accuracy of ARI data.

Then we analyze the density of data integration. The integration effect on CA data is as follows: The integration curve  $L_{CA}$  integrated with  $(Average_X = 0.4911, Average_Y = 0.0851)$  as the center. The entanglement error is  $\omega = \frac{Average_Y}{Average_X} = 0.17$  and the integration curve radius is  $R_{CA} \doteq 1.5$ . The data integration density is higher than before, and the imbalanced data that presents the red part and the balanced data that presents the black part are completely separated, which fully shows that the DE-MPFFC algorithm has a better distribution advantage for improving the accuracy of CA data integration. The integration effect on NMI data is as follows: The integration curve  $L_{NMI}$  integrated with  $(Average_X = 0.2864, Average_Y = -0.0671)$  as the center. The entanglement error is  $\omega = \frac{Average_Y}{Average_X} = -0.23$  and the integration curve radius is  $R_{NMI} \doteq 1.6$ . The data integration density is higher than before, and the imbalanced data that presents the red part and the balanced data that presents the black part are completely separated, which fully shows that the DE-MPFFC algorithm has a better distribution advantage for improving the accuracy of NMI data integration. The integration effect on ARI data is as follows: The integration curve  $L_{ARI}$  integrated with  $(Average_X = 0.4510,$

$Average_Y = -0.0283$ ) as the center. The entanglement error is  $\omega = \frac{Average_Y}{Average_X} = -0.06$  and the integration curve radius is  $R_{ARI} \doteq 1.5$ . The data integration density is higher than before and the imbalanced data that presents the red part and the balanced data that presents the black part are completely separated, which fully shows that the DE-MPFSC algorithm has a better distribution advantage for improving the accuracy of ARI data integration.

Finally, we analyze the degree of data separation. From the distribution of Figures 16–18, CA data, NMI data and ARI data have been completely separated by the DE-MPFSC algorithm. The data integration effect presents the following trend: the data set that is balanced is in the black curve part and tends to converge toward the center of the integration curve, and the imbalanced data set is in the red curve part and tends to converge toward the center of the integration curve.

The above analysis shows that the DE-MPFSC algorithm has a good natural advantage in integrating imbalanced data.

## 7. Conclusions

In this paper, some processing problems of imbalanced data. We design the algorithm based on the DE algorithm, constructs a differential evolution algorithm based on a mixed penalty function screening criterion for imbalanced data integration. The method can improve the classification problem of imbalanced data and construct an empirical analysis through UCI machine learning datasets. In addition, the verification result is in accordance with the imbalanced data classification expectation and the integration effect. The main work of this article is as follows:

1. Based on the normalization of internal and external penalty functions, we established the mixed penalty function screening criteria on DE algorithm.
2. We established a differential evolution integration algorithm (DE-MPFSC algorithm) based on the mixed penalty function screening criterion, which broadens the algorithm foundation for the efficient integration of imbalanced data.
3. We constructed the Markov process of the DE-MPFSC algorithm and proved the theoretical validity and evolution mechanism of DE-MPFSC algorithm.
4. We creatively introduce the entanglement and entanglement error and compare the performance of the DE-MPFSC algorithm through data analysis.
5. Based on the empirical analysis of the UCI machine learning data, we further illustrated the effectiveness of the DE-MPFSC algorithm in the efficient integration of the imbalanced data, leading to multimode imbalanced data integration.

**Author Contributions:** Y.G. provided the research ideas of the paper; K.W. solved the overall framework, algorithm framework, paper structure, and result analysis of the research ideas of the paper; C.G. provides data analysis and corresponding analysis results; Y.S. provides the overall research ideas and main methods of the paper; T.L. has helped the writing of the paper.

**Funding:** This work was supported by the Major Scientific Research Special Projects of North Minzu University (No.ZDZX201901), the NSFC (No.61561001;11961001), Graduate Innovation Project of North Minzu University (No:YCX19120), First-Class Disciplines Foundation of Ningxia (No:NXYLXK2017B09) and the by National Natural Science Foundation of Shanxi Province (2019JM-425), China Postdoctoral Science Foundation Funded Project (2019M653567).

**Acknowledgments:** Acknowledgment for the financial support of the National Natural Science Foundation Project, the University-level Project of Northern University for Nationalities and the District-level Project of Ningxia, the Major Scientific Research Special Projects of North Minzu University and all authors' efforts. And the reviewers and instructors for the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Everitt, B. Cluster Analysis. *Qual. Quant.* **1980**, *14*, 75–100. [[CrossRef](#)]
2. Chunyue, S.; Zhihuan, S.; Ping, L.; Wenyuan, S. The study of Naive Bayes algorithm online in data mining. In Proceedings of the World Congress on Intelligent Control and Automation, Hangzhou, China, 15–19 June 2004.
3. Samanta, B.; Al-Balushi, K.R.; Al-Araimi, S.A. Artificial neural networks and genetic algorithm for bearing fault detection. *Soft Comput. Fusion Found. Methodol. Appl.* **2006**, *10*, 264–271. [[CrossRef](#)]
4. Díez-Pastor, J.F.; Rodríguez, J.J.; García-Osorio, C.; Kuncheva, L.I. Random Balance: Ensembles of variable priors classifiers for imbalanced data. *Knowl.-Based Syst.* **2015**, *85*, 96–111. [[CrossRef](#)]
5. Maldonado, S.; Weber, R.; Famili, F. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Inf. Sci.* **2014**, *286*, 228–246. [[CrossRef](#)]
6. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
7. Vorraboot, P.; Rasmeequan, S.; Chinnasarn, K.; Lursinsap, C. Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms. *Neurocomputing* **2015**, *152*, 429–443. [[CrossRef](#)]
8. Krawczyk, B.; Woźniak, M.; Schaefer, G. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl. Soft Comput.* **2014**, *14*, 554–562. [[CrossRef](#)]
9. López, V.; Río, S.D.; Benítez, J.M.; Herrera, F. Cost-sensitive linguistic fuzzy rule-based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets Syst.* **2015**, *258*, 5–38. [[CrossRef](#)]
10. Santucci, V.; Milani, A.; Caraffini, F. An Optimisation-Driven Prediction Method for Automated Diagnosis and Prognosis. *Mathematics* **2019**, *7*, 1051. [[CrossRef](#)]
11. Mikalef, P.; Boura, M.; Lekakos, G.; Krogstie, J. Big data analytics and firm performance: Findings from a mixed-method approach. *J. Bus. Res.* **2019**, *98*, 261–276. [[CrossRef](#)]
12. Thai-Nghe, N.; Gantner, Z.; Schmidt-Thieme, L. Cost-sensitive learning methods for imbalanced data. In Proceedings of the International Joint Conference on Neural Networks, Barcelona, Spain, 18–23 July 2010; pp. 1–8.
13. Freund, Y. Boosting a weak learning algorithm by majority. *Inf. Comput.* **1995**, *121*, 256–285. [[CrossRef](#)]
14. Breiman, L. Bagging predictors, machine learning research: Four current directors. *ResearchGate* **1996**, *24*, 123–140.
15. Sun, Z.; Song, Q.; Zhu, X.; Sun, H.; Xu, B.; Zhou, Y. A novel ensemble method for classifying imbalanced data. *Pattern Recognit.* **2015**, *48*, 1623–1637. [[CrossRef](#)]
16. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. Knowledge Discovery in Databases: Pkdd 2003. In Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, 22–26 September 2003; pp. 107–119.
17. Flach, P.A.; Lachiche, N. Naive Bayesian Classification of Structured Data. *Mach. Learn.* **2004**, *57*, 233–269. [[CrossRef](#)]
18. Fernández, A.; López, V.; Galar, M.; Del Jesus, M.; Herrera, F. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowl.-Based Syst.* **2013**, *42*, 97–110. [[CrossRef](#)]
19. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
20. Wang, K.; Gao, Y. Topology Structure Implied in  $\beta$  – Hilbert Space, Heisenberg Uncertainty Quantum Characteristics and Numerical Simulation of the DE Algorithm. *Mathematics* **2019**, *7*, 330. [[CrossRef](#)]
21. Das, S.; Mullick, S.S.; Suganthan, P.N. Recent advances in differential evolution—An updated survey. *Swarm Evol. Comput.* **2016**. [[CrossRef](#)]
22. Neri, F.; Tirronen, V. Recent advances in differential evolution: A survey and experimental analysis. *Artif. Intell. Rev.* **2010**, *33*, 61–106. [[CrossRef](#)]
23. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [[CrossRef](#)]

24. Brest, J.; Zumer, V.; Maucec, M.S. Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver, BC, Canada, 16–21 July 2006.
25. Brest, J.; Zamuda, A.; Bošković, B.; Greiner, S.; Zumer, V. An Analysis of the Control Parameters' Adaptation in DE. *Stud. Comput. Intell.* **2008**, *3*. [[CrossRef](#)]
26. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M.A. Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **2008**, *12*, 64–79. [[CrossRef](#)]
27. Das, S.; Abraham, A.; Chakraborty, U.K.; Konar, A. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Trans. Evol. Comput.* **2009**, *13*, 526–553. [[CrossRef](#)]
28. Mallipeddi, R.; Suganthan, P.N. Ensemble of Constraint Handling Techniques. *IEEE Trans. Evol. Comput.* **2010**, *14*, 561–579. [[CrossRef](#)]
29. Qu, B.Y.; Suganthan, P.N. Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods. *Eng. Optim.* **2011**, *43*, 403–416. [[CrossRef](#)]
30. Qin, A.K.; Suganthan, P.N. Self-adaptive differential evolution algorithm for numerical optimization. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005), Edinburgh, UK, 2–4 September 2005.
31. Zou, D.; Liu, H.; Gao, L.; Li, S. A modified differential evolution algorithm for unconstrained optimization problems. *Neurocomputing* **2011**, *120*, 1608–1623. [[CrossRef](#)]
32. Ghosh, A.; Das, S.; Chowdhury, A.; Giri, R. An improved differential evolution algorithm with fitness-based adaptation of the control parameters. *Inf. Sci.* **2011**, *181*, 3749–3765. [[CrossRef](#)]
33. Caraffini, F.; Kononova, A.V.; Corne, D. Infeasibility and structural bias in Differential Evolution. *Inf. Sci.* **2019**, *496*, 161–179. [[CrossRef](#)]
34. Storn, R. System design by constraint adaptation and differential evolution. *IEEE Trans. Evol. Comput.* **1999**, *3*, 22–34. [[CrossRef](#)]
35. Thomsen, R. Flexible ligand docking using differential evolution. In Proceedings of the 2003 Congress on Evolutionary Computation (CEC '03), Canberra, ACT, Australia, 8–12 December 2003; pp. 2354–2361; Volume 4.
36. Ali, M.; Pant, M.; Singh, V.P. An improved differential evolution algorithm for real parameter optimization problems. *Int. J. Recent Trends Eng.* **2009**, *1*, 63–65.
37. Yang, M.; Li, C.; Cai, Z.; Guan, J. Differential evolution with auto-enhanced population diversity. *IEEE Trans. Cybern.* **2015**, *45*, 302–315. [[CrossRef](#)] [[PubMed](#)]
38. Iri, M.; Imai, H. Theory of the multiplicative penalty function method for linear programming. *Discret. Algorithms Complex.* **1987**, *30*, 417–435.
39. Rao, S.S.; Bard, J. *Engineering Optimization: Theory and Practice*, 4th ed.; A I I E Transactions; John Wiley & Sons: New York, NY, USA, 1997; Volume 29; pp. 802–803.
40. Wright, M. The interior-point revolution in optimization: History, recent developments, and lasting consequences. *Bull. Am. Math. Soc.* **2005**, *42*, 39–57. [[CrossRef](#)]
41. Deb, K. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 311–338. [[CrossRef](#)]
42. Venkatraman, S.; Yen, G.G. *A Generic Framework for Constrained Optimization Using Genetic Algorithms*; IEEE Press: Piscataway, NJ, USA, 2005.
43. Xie, X.F.; Zhang, W.J.; Bi, D.C. Handling equality constraints by adaptive relaxing rule for swarm algorithms. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 19–23 June 2004; Volume 2.
44. Schwefel, H.P. *Evolution and Optimum Seeking*; Wiley: New York, NY, USA, 1995.
45. Kramer, O. Premature Convergence in Constrained Continuous Search Spaces. In Proceedings of the Parallel Problem Solving from Nature—PPSN X, Dortmund, Germany, 13–17 September 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 62–71.
46. Gasparini, M. *Markov Chain Monte Carlo in Practice*; Markov chain monte carlo in practice; Chapman and Hall: London, UK, 1996; pp. 9236–9240.
47. Eiben, A.E.; Aarts, E.H.L.; Hee, K.M.V. *Global Convergence of Genetic Algorithms: A Markov Chain Analysis*; Parallel Problem Solving from Nature; Springer: Berlin/Heidelberg, Germany, 1991; pp. 3–12.

48. Rudolph, G. Convergence of evolutionary algorithms in general search spaces. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 50–54.
49. Cerf, R. Asymptotic Convergence of Genetic Algorithms. *Adv. Appl. Probab.* **1998**, *30*, 521–550. [[CrossRef](#)]
50. Xu, Z.B.; Nie, Z.K.; Zhang, W.X. Almost sure strong convergence of a class of genetic algorithms with parent-offspring competition. *Acta Math. Appl. Sin.* **2002**, *25*, 167–175.
51. Unified Communications Irvine Machine Learning Repository(UCI). Available online: <https://archive.ics.uci.edu/ml/index.php> (accessed on 1 August 2019).
52. Silva, L.O.; Zárate, L.E. A brief review of the main approaches for treatment of missing data. *Intell. Data Anal.* **2014**, *18*, 1177–1198. [[CrossRef](#)]
53. Batista, G.A.P.A. MariaCarolinaMonard. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* **2003**, *17*, 519–533. [[CrossRef](#)]
54. Liang, J.; Bai, L.; Dang, C.; Cao, F. The K-Means-Type Algorithms Versus Imbalanced Data Distributions. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 728–745. [[CrossRef](#)]
55. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
56. Zhao, X.; Liang, J.; Dang, C. Clustering ensemble selection for categorical data based on internal validity indices. *Pattern Recognit.* **2017**, *69*, 150–168. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).