Applied Soft Computing Journal 100 (2021) 106724

Contents lists available at ScienceDirect

Applied Soft Computing Journal

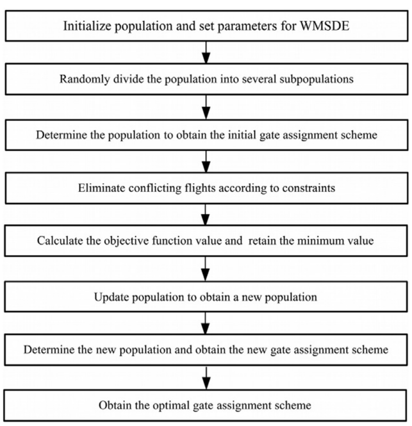
journal homepage: www.elsevier.com/locate/asoc

Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem

Wu Deng [a,c], Junjie Xu [a], Yingjie Song [b], Huimin Zhao [a,d,*]

[a] College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China
[b] Co-innovation Center of Shandong Colleges and Universities: Future Intelligent Computing, Shandong Technology and Business University, Yantai 264005, China
[c] Traction Power State Key Laboratory of Southwest Jiaotong University, Chengdu 610031, China
[d] The State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 31 January 2020
Received in revised form 29 July 2020
Accepted 13 September 2020
Available online 15 September 2020

Keywords:
Differential evolution
Wavelet basis function
Optimal mutation strategy
Optimization and control
Gate assignment

ABSTRACT

The optimization performance of differential evolution(DE) algorithm significantly depends on control parameters and mutation strategy. However, it is difficult to set suitable control parameters and select reasonable mutation strategy for DE in solving an actual engineering optimization problem. To solve these problems, a new optimal mutation strategy based on the complementary advantages of five mutation strategies is designed to develop a novel improved DE algorithm with the wavelet basis function, named WMSDE, which can improve the search quality, accelerate convergence and avoid fall into local optimum and stagnation. In the proposed WMSDE, the initial population is divided into several subpopulations to exchange search information between the different subpopulations and improve the population diversity to a certain extent. The wavelet basis function and normal distribution function are used to control the scaling factor and the crossover rate respectively in order to ensure the diversity of solutions and accelerate convergence. The new optimal mutation strategy is used to improve the local search ability and ensure the global search ability. Finally, the proposed WMSDE is compared with five state-of-the-art DE variants by 11 benchmark functions. The experiment results indicate that the proposed WMSDE can avoid premature convergence, balance local search ability and global search ability, accelerate convergence, improve the population diversity and the search quality. Additionally, a real-world airport gate assignment problem is employed to further prove the effectiveness of the proposed WMSDE. The results show that it can effectively solve the complex airport gate assignment problem, and obtain airport gate assignment rate of 97.6%.

1. Introduction

A lot of problems in engineering application and scientific research can be translated directly into the optimization problems [1–3]. These optimization problems belong to NP-Hard

* Corresponding author at: College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China.
E-mail address: hm_zhao1977@126.com (H. Zhao).

https://doi.org/10.1016/j.asoc.2020.106724
1568-4946/

problems, which are difficult to be solved by traditional methods [4–6]. In the past decades, a lot of researchers have proposed some intelligent optimization methods to solve the NP-Hard problems, such as genetic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO), ant colony optimization (ACO), fish swarm optimization (FSO), island artificial bee colony (IABC), Island flower pollination (IFP), Island bat algorithm (IBA), grey wolf optimizer (GWO), bacterial foraging optimization (BFO), and so on [7–15]. Although these intelligent optimization methods can better solve the NP-Hard problems and obtain better solution than traditional methods, e.g. linear programming method, numerical probability algorithm and so on, they still exist lower accuracy and more computation time. At the same time, with the increasing complexity of optimization problems, the studies of the optimization methods have become popular topics [16–18].

Differential Evolution (DE) algorithm is a random evolution algorithm based on population evolution proposed by Storn and Price [19]. Because the principle of DE is simple, and easy to understand and implement, it has stronger robustness and search ability, and fewer control parameters. But it is prone to premature, localized optimality and low convergence in the late stage of search. To solve these problems, many scholars have done a lot of works to improve the optimization performance of the DE in recent years. Su and Lee [20] proposed an improved mixed-integer hybrid DE method. Das et al. [21] proposed an improved DE for large unlabeled data. Lai and Cao [22] proposed an improved DE to solve the vehicle routing problem. Dorronsoro and Bouvry [23] proposed several DE variants using different panmictic and decentralized population schemes. Elsayed et al. [24] proposed an improved DE that uses a mix of different mutation operators. Jia et al. [25] proposed an improved version of (mu+lambda)-CDE. Gong and Cai [26] proposed an improved multi-strategy adaptive DE algorithm. Tang et al. [27] proposed an improved DE to solve practical dynamic scheduling. Zhang et al. [28] proposed an improved constrained DE algorithm. Mohamed [29] proposed an improved DE to solve global numerical optimization problems. Yi et al. [30] proposed a novel DE algorithm by implementing pbest roulette wheel selection and retention mechanism. Guo et al. [31] proposed an improved constraint-activated DE algorithm. Tian et al. [32] proposed a novel DE algorithm based on improved individual-based parameter setting and diversity-based selection strategy. Cai et al. [33] proposed a cooperative coevolution DE algorithm. Awad et al. [34] proposed an improved version of multi-objective DE algorithm. Ho-Huu et al. [35] proposed a novel DE to solve the shape and size optimization problems. Maucec et al. [36] proposed a new variation of DE for large-scale black-box optimization. Wang et al. [37] proposed a self-adaptive DE algorithm with improved mutation strategy. Ajithapriyadarsini et al. [38] proposed an adaptive fuzzy logic-based DE algorithm. Yazdani and Hadavandi [39] proposed a linearized monarch butterfly optimization algorithm improved with DE. Vafashoar and Meybodi [40] proposed a multi-population DE algorithm using cellular learning automata and evolutionary context information. Wang et al. [41] proposed a self-adaptive ensemble-based DE. Ben [42] proposed an accelerated DE algorithm with new operators. Deng et al. [43] proposed an improved quantum-inspired differential evolution algorithm.

Through the survey and analysis of these literatures, we know that the new mutation operators, the strategies of adaptive control parameters, multi-strategy and multi-population and so on are proposed to improve the optimization performance of the DE for solving optimization problems in the last few decades. Although these improved DE algorithms have been achieved better optimization effects in solving optimization problem, it is still easy to fall into local optimum, has poor optimization ability and low convergence. Therefore, in order to solve these defects, a new optimal mutation strategy based on the complementary advantages of the five mutation strategies is designed, then a novel improved DE algorithm based on the wavelet basis function and new optimal mutation strategy, named WMSDE is proposed to improve the optimization performance of the DE, avoid to fall into local optimum and increase convergence. In the proposed WMSDE, the initial population is divided into several subpopulations according to the solving complex optimization problem in order to exchange search information between the different subpopulations and improve the population diversity to a certain extent. The wavelet basis function and normal distribution function are introduced to control the scaling factor and the crossover rate respectively in order to ensure the diversity of solutions, accelerate convergence and improve optimization performance. A new optimal mutation strategy based on the complementary advantages of five mutation strategies in the first generation is designed to improve the local search ability and ensure the global search ability. 11 benchmark functions and a real-world airport gate assignment problem are employed to validate the effectiveness of the proposed WMSDE algorithm. The experiment results show that the proposed WMSDE algorithm can effectively avoid premature convergence, balance local search ability and global search ability, and also obtain ideal airport gate assignment results. The gate assignment rate reaches at 97.6%.

The research presented in this paper is organized as follows: Section 2 introduced the DE algorithm. In Section 3, the proposed WMSDE algorithm is described and the contribution is highlighted. In Section 4, the numerical experiments and analysis to solve benchmark functions are provided. In Section 5, airport gate assignment method is proposed and described in detail. The experiments and results to assign airport gates are provided to validate the effectiveness of the proposed WMSDE algorithm in Section 6. The Section 7 is given to conclude the WMSDE algorithm and suggest some works in the future.

2. Differential evolution algorithm

The DE algorithm uses the difference between individuals to guide this algorithm to search in the solution space [19]. It mainly includes initialization population, mutation operation, crossover operation, selection operation, and so on. The main idea of the DE is to differentiate and scale between two different individual vectors in the same population, and add a third individual vector in this population to obtain a mutation individual vector, which is crossed with the parent individual vector with a certain probability to generate an attempted individual vector. Finally, the attempted individual vector and the parent individual vector are executed greedy selection, and the better individual vector is saved to the next generation. The basic evolution processes of the DE are described as follows.

2.1. Initialization

The DE algorithm uses D-dimensional vectors(M) as the initial solution. Set population number(N), each individual can be expressed as $x_i(G) = (x_{i1}(G), x_{i2}(G), \ldots, x_{iD}(G))$. The initial population is generated in $[x_{min}, x_{max}]$. In here, M is the number of D-dimensional vectors, N is the number of populations, and $x_i(G)$ is the ith individual.

$$x_{iD} = x_{min} + rand(0, 1) * (x_{max} + x_{min}) \tag{1}$$

where G represents the Gth generation, x_{max} represents the maximum value of the search space, x_{min} represents the minimum value of the search space, rand(0, 1) represents a random number that meets a normal distribution within (0,1).

2.2. Mutation operation

The DE algorithm uses the mutation operation to generate a mutation vector $V_{i,G}$ for each individual $x_{i,G}$ in the current population (target vector). For each generated target vector, a corresponding mutation vector can be generated by a certain mutation strategy. According to the different generation methods of mutation individuals, several different mutation strategies for the DE are formed. The five most commonly used mutation strategies are described as follows.

(1) DE/rand/1

$$V_{i,G} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \tag{2}$$

(2) DE/best/1

$$V_{i,G} = x_{best,G} + F \cdot (X_{r_1,G} - x_{r_2,G}) \tag{3}$$

(3) DE/rand-to-best/1

$$V_{i,G} = x_{i,G} + F \cdot \left(x_{best,G} - x_{i,G}\right) + F \cdot (x_{r_1,G} - x_{r_2,G}) \tag{4}$$

(4) DE/best/2

$$V_{i,G} = x_{i,G} + F \cdot \left(x_{best,G} - x_{i,G}\right) + F \cdot (x_{r_1,G} - x_{r_2,G}) \tag{5}$$

(5) DE/rand/2

$$V_{i,G} = x_{r_1,G} + F \cdot \left(x_{r_2,G} - x_{r_3,G}\right) + F \cdot (x_{r_4,G} - x_{r_5,G}) \tag{6}$$

where, r_1, r_2, r_3, r_4 and r_5 are randomly generated exclusive integers within $[1, M]$. The scaling factor F is a positive control parameter to scale the difference vector. $x_{best,G}$ is the best individual vector with the best fitness value in the Gth generation.

2.3. Crossover operation

Each pair of target vectors $x_{i,G}$ and their corresponding mutation vectors $V_{i,G}$ are crossed to generate a test vector $U_{i,G} = (u_{1,G}, u_{2,G}, \ldots, u_{i,G})$. In the DE algorithm, a binomial crossover is defined as follow.

$$u_{i,G} = \begin{cases} v_{i,G} & if \left(rand_j\,(0,1) \le CR\right)\; or \;(j = j_{rand}, j = 1, 2, 3, \ldots, D) \\ x_{i,G} & otherwise \end{cases} \tag{7}$$

where, the crossover rate CR is a specified constant on [0,1], which is used to control the duplicated proportion from mutation vector. j_{rand} is a randomly selected integer on $[1, D]$.

2.4. Selection operation

If the values of parameters exceed the corresponding upper or lower bounds, they can be reinitialized randomly and uniformly within the given range. Then the objective function values of all test vectors are evaluated, and the selection operation is performed. The objective function value $f(U_{i,G})$ of each test vector is compared with the objective function value of the corresponding target vector in the current population. If the objective function value of the test vector is less than or equal to that of the corresponding target vector, then the target vector is replaced by the test vector for the next generation. Otherwise, the target vector is remained for the next generation. The selection operation can be expressed as follow.

$$X_{i,G+1} = \begin{cases} U_{i,G} & if\; (f(U_{i,G}) \le f(X_{i,G})) \\ X_{i,G} & otherwise \end{cases} \tag{8}$$

3. An improved differential evolution (WMSDE) algorithm

The optimization performance of the DE is highly dependent on the evolution strategies and the associated parameter values. The DE mainly includes the scaling factor F and the crossover rate CR, which have a great influence on the optimization performance. The scaling factor F is closely related to the search step size. When the population is far from the global best value, the larger search step size can help to quickly converge to a better subspace. When the population is closer to the global best value, the smaller search step size can help to accurately search better solutions. The crossover rate CR reflects the probability that the offspring directly inherit information from their parents. It has a great influence on the search ability and convergence, and can effectively improve the optimization performance of the DE algorithm. However, the control parameters need to be set by experiment or experience, which result in lower efficiency and reliability. When the mutation strategy is selected, the greedy strategy is fast, but it is easy to fall into local optimum value. The population diversity strategy has strong search ability, but the efficiency is often low. The inappropriate strategy and parameters for the DE may lead to premature convergence or stagnation. Therefore, in order to solve these problems, on the basis of the characteristics of the five mutation strategies of the DE algorithm, a new optimal mutation strategy based on complementary advantages of the five mutation strategies in the first generation is designed, which is used as the mutation strategy for DE in the subsequent iterations in order to improve the local search ability and ensure the global search characteristics. Then the wavelet basis function is introduced into the DE with new optimal mutation strategy to propose an improved differential evolution algorithm, named WMSDE in this paper. In the proposed WMSDE algorithm, the initial population is divided into several subpopulations according to the solved complex optimization problem in order to exchange search information between the different subpopulations and improve the population diversity. The wavelet basis function and the normal distribution are introduced into the DE to effectively control the scaling factor F and the crossover rate CR respectively in order to ensure the solution diversity, accelerate the convergence speed and improve the optimization performance.

3.1. Multi-population strategy

Multi-population strategy is to divide the population into several subpopulations to realize the information exchange among the various subpopulations. In each iteration, the parallel evolution mechanism is used to dynamically balance the global ability and local search ability, improve the convergence speed. Because each subpopulation is a subspace in the solution space, the search strategy is used to update each subpopulation. The individual with the best fitness value is migrated among the different subpopulations to complete the exchange information among these individuals and effectively improve the searching efficiency. Finally, a new population is generated from these subpopulations. Therefore, the multi-population strategy can avoid the premature convergence in the evolution process, and takes on stronger global and local search ability, accelerates the population search speed and improves the convergence accuracy in solving complex optimization problems.

3.2. Improved scaling factor

The scaling factor F in the DE is closely related to the search step size, and the DE has different search step sizes in different search stages. The scaling factor F is a constant value in the

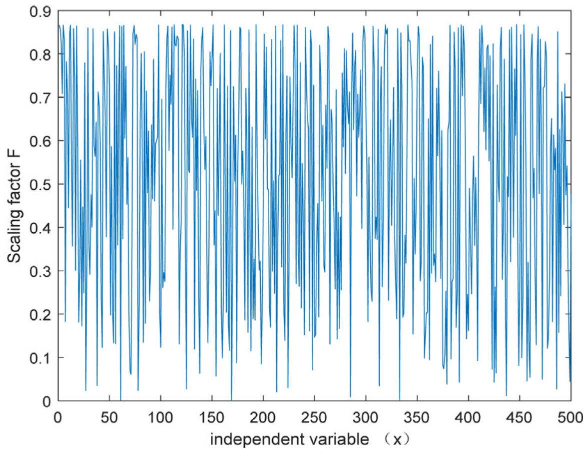


Fig. 1. The value of the scaling factor F.

basic DE, the value of the scaling factor F is generally selected within [0,2], usually 0.5. The value of the scaling factor F seriously affect the optimization performance of the DE algorithm. Wavelet basis function is a series of functions obtained from the expansion and translation. Therefore, the wavelet basis function is introduced into the DE in order to make full use of the characteristics of the wavelet basis function to achieve the improvement of the scaling factor F. Because the Mexican Hat (mexh) wavelet function is the second derivative of the Gauss function, it has good localization in the time domain and the frequency domain. Therefore, the Mexican Hat (mexh) wavelet is used to improve the scaling factor F. The value of Mexican Hat (mexh) wavelet is random value between 0 and 1, which can avoid premature convergence and fall into local optimum, and also improve the solution diversity. The improved scaling factor F is described as follow.

$$F = \frac{2}{\sqrt{3}} \cdot \pi^{-\frac{1}{4}} \cdot (1 - x^2) \cdot e^{-\frac{x^2}{2}} \tag{9}$$

According to the characteristics of wavelet basis function, the variable in each generation is x, which is a random number within (0,1). Therefore, the value of scaling factor F is a random value according to the variable x. Therefore, the value of scaling factor F is within $[0, \frac{2}{\sqrt{3}} \cdot \pi^{-\frac{1}{4}}]$. The wavelet basis function can better describe the characteristics of the scaling factor F.

The value of the scaling factor F is shown in Fig. 1.

3.3. A new optimal mutation strategy

The mutation strategy can be simplified as a weighted expression between the basic vector and difference vector. The basic vector is used to guide and adjust the evolution direction of the population, and the difference vector plays the roles of random disturbance and fine search. The mutation strategy can maintain the balance between the diversity and convergence to some extent. The five commonly used mutation strategies have their own characteristics. DE/rand/1 strategy can better deal with single-peak and multi-peak optimization problem, but the convergence is poor. DE/rand/2 strategy has better global search ability, but the convergence speed is slow. DE/best/1 and DE/best/2 strategies have a fast convergence rate, but their global exploration abilities are relatively weak and they are easy to fall into local convergence. DE/rand-to-best/1 strategy has relatively balanced global exploration and local optimization, but the robustness is relatively poor. For different complex optimization problems, each mutation strategy has different optimization ability, and the obtained results are also different. Therefore, in order to make full use of ability of different mutation strategy for solving complex optimization problems, a new optimal mutation strategy based on complementary advantages of the five mutation strategies is designed in this paper. Firstly, the DE algorithm with five different mutation strategies makes a test for solving the complex optimization problem in in the first iteration, respectively. Then the best one based on this test is chosen as the optimal mutation strategy for DE algorithm in the subsequent iterations until the convergence is achieved or the maximum number of iterations is reached. We define the mutation strategy as a new optimal mutation strategy for DE algorithm, which can improve the local search ability and ensure the global search ability.

3.4. Improved crossover rate CR

The crossover rate CR reflects the probability that the offspring directly inherit information from their parents, it has a great influence on the search ability and convergence speed. The larger value of the CR can make the offspring to depend on the mutation process and inherit less information from the parent generation. In this strategy, a wide range of global search can be realized and the possibility can be improved to jump out from local optimum. On the contrary, if the offspring locally searches around the parent generation, the smaller value of the CR is set to accelerate the convergence and improve the solution accuracy. Therefore, a uniform distribution strategy is used to improve the crossover rate CR. The expression is described as follow.

$$CR = Norm\,(0, 1) \tag{10}$$

The crossover rate CR is set as a random number to increase the diversity. When the DE algorithm is not faced with the problem of a prior knowledge, it can still automatically generate an appropriate value of parameters in the current search, which can improve the optimization performance.

3.5. Model and steps of the WMSDE algorithm

The flow of the WMSDE algorithm based on wavelet basis function and new optimal mutation strategy is shown in Fig. 2.

The detailed steps of the WMSDE algorithm are described as follows.

Step 1. The initial population $X_i = (x_1, x_2, \ldots, x_i)$ is randomly generated within (x_{max}, x_{min}). Initialize all parameters, including population size (NP), the maximum number of iterations (G), the number of initial iteration ($G = 1$).

Step 2. If the DE algorithm is used to solve the complex optimization problems in the first generation, then Step 3 is executed. Otherwise, Step 5 is executed.

Step 3. Five different mutation vectors are respectively generated by using five different mutation strategies, and the fitness values are calculated and obtained by using DE with five different mutation strategies in the first generation.

Step 4. The obtained fitness values are compared in order to select the best one based on this test, which is called a new optimal mutation strategy.

Step 5. The DE algorithm with the new optimal mutation strategy is used to solve the complex optimization problems in the subsequent iterations.

Step 6. Execute the crossover operation, and generate a random number within (0,1), then compare with the crossover rate CR.

Step 7. If the generated value of the crossover operation is smaller than the value of the CR, the generated mutation vector is selected as the test vector. Otherwise, the contemporary target vector is selected as the test vector.

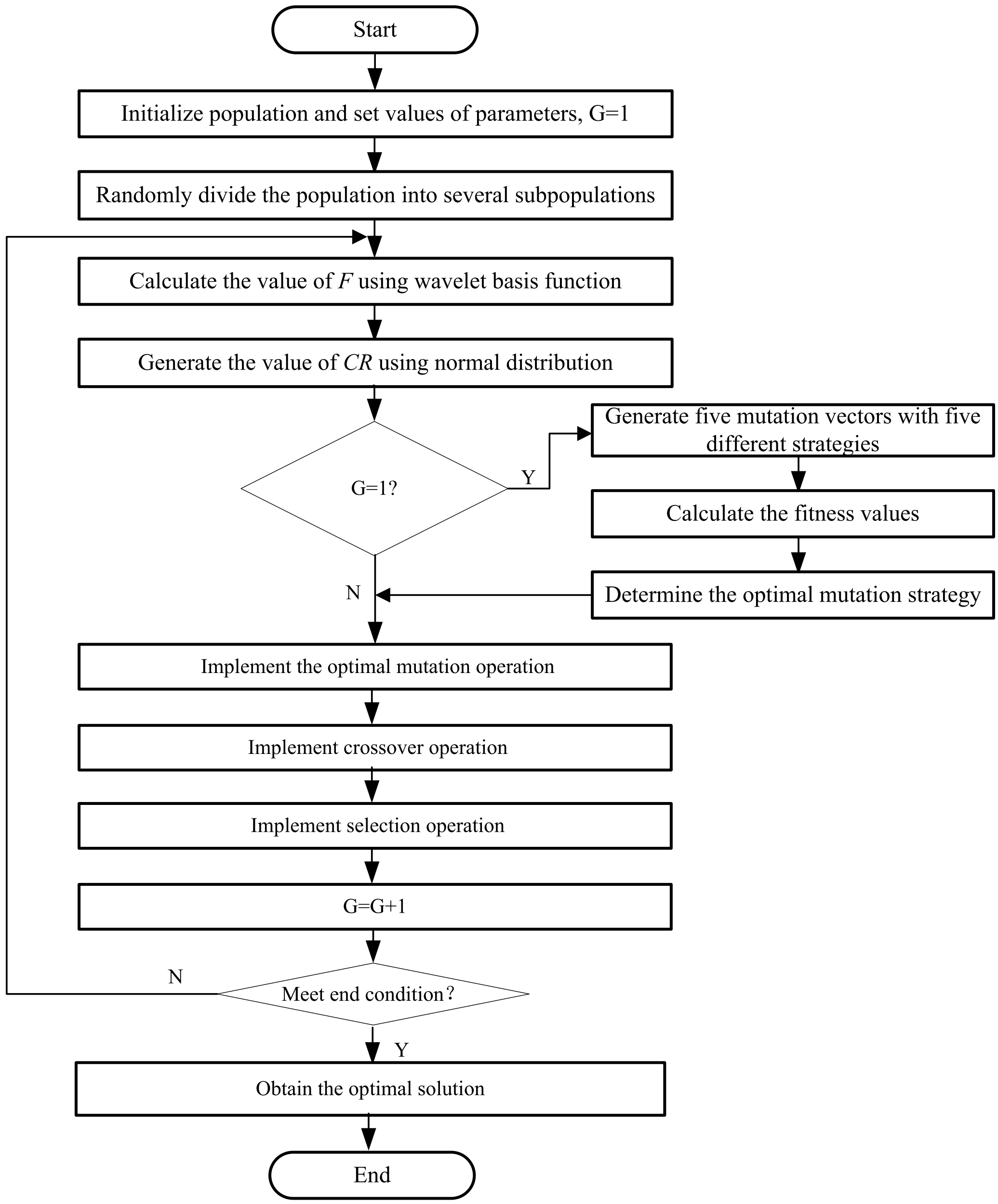


Fig. 2. The flow of the WMSDE algorithm.

Step 8. Execute the selection operation, calculate the fitness value of the target vector and the test vector, and compare and select the individuals with the optimal fitness value for the next generation.

Step 9. If the maximum number of iterations is reached or the error requirement is met, the WMSDE algorithm is end. And the optimal value in solving complex optimization problem is output. Otherwise skip to Step 5.

4. Numerical experiments and analysis

4.1. Test functions

In order to evaluate the optimization performance of the proposed WMSDE algorithm, 11 benchmark functions are selected in here. The expressions, value ranges and minimum values of 11 benchmark functions are shown in Table 1. D is the number of dimensions. For the 11 benchmark functions, the benchmark functions $f_1 \sim f_4$, f_6 and f_7 are single peak functions, which are mainly used to evaluate the accuracy and convergence speed. $f_8 \sim f_{11}$ are multimodal functions, which are mainly used to evaluate the global search stability. The initial parameters of the WMSDE algorithm is selected after thorough testing. In the simulation experiments, the alternative values were tested and modified for some functions to obtain the most reasonable initial values of these parameters. These selected values of the parameters take on the optimal solution and the most reasonable running time of these algorithms to efficiently complete the problem solving. Therefore, according to the simulation experiment results and related references, the selected values of these parameters and five mutation strategies of the WMSDE algorithm are shown in Table 2.

4.2. Experimental results and analysis

4.2.1. Experimental results

The obtained experiment results of 11 benchmark functions are shown in Table 3. In here, the Best represents the obtained

Table 1
Benchmark functions.

Functions	Range	Optimal value
$f_1(x)=\sum_{i=1}^{D} x_i^2$	[−100,100]	0
$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[−10,10]	0
$f_3(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	[−100,100]	0
$f_4(x)=max_i\{\lvert x_i\rvert\}$	[−100,100]	0
$f_5(x)=\sum_{i=1}^{D-1}[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2]$	[−30,30]	0
$f_6(x)=\sum_{i=1}^{D}\lfloor x_i+0.5\rfloor^2$	[−100,100]	0
$f_7(x)=\sum_{i=1}^{D} ix_i^4+rand[0,1)$	[−1.28,1.28]	0
$f_8(x)=\sum_{i=1}^{D}-x_i\sin\sqrt{x_i}$	[−500,500]	−12 569.5
$f_9(x)=\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[−5.12,5.12]	0
$f_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	[−32,32]	0
$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[−600,600]	0

Table 2
The five mutation strategies and initial parameters of the WMSDE algorithm.

Algorithms	Strategies	Population size	Dimensions	Iterations	Running times
DE	DE/rand/1	100	30/50	2000	30
	DE/best/1	100	30/50	2000	30
	DE/rand-to/best/1	100	30/50	2000	30
	DE/best/2	100	30/50	2000	30
	DE/rand/2	100	30/50	2000	30
WMSDE	New optimal mutation strategy	100	30/50	2000	30

optimal value in the results of 30 times, Mean represents the average value of 30 times, and Std represents the standard deviation of 30 times.

As can be seen from Table 3, for the function $f_1 \sim f_3$, the WMSDE algorithm can obtain the optimal value, mean and standard deviation in both 30 and 50 dimensions, and it can obtain the best optimal value(zero) for the functions f_1 and f_2. For the functions f_4 and f_6, the DE algorithm with five different mutation strategies and WMSDE algorithm can obtain the best optimal value(zero). For the function f_5, the optimization performance of all algorithms is not good, because this function is a single peak function in the independent variable 2∼3 dimensions. But the obtained results of the WMSDE algorithm is best than the DE algorithm with five different mutation strategies. For the function f_7, the WMSDE algorithm shows better optimization ability in the optimal value and average value than other several DE algorithms. For the function f_8, compared with the DE algorithm with five different mutation strategies, the WMSDE algorithm performs well in the mean and optimal values, and obtains the best optimal value. This experiment result shows that the WMSDE algorithm is very stable. For the functions f_9 and f_{11}, the DE algorithms with mutation strategies DE/rand/1 and DE/rand/2 cannot obtain better optimization values, but the WMSDE algorithm can obtain best optimal value(zero). For the function f_{10}, although the DE algorithm with five different mutation strategies can obtain relatively optimal value, these algorithms do not obtain the optimal value(zero). But the WMSDE algorithm can obtain the relative optimal value and standard deviation(zero), which shows the optimization ability, stability, convergence and diversity of the WMSDE algorithm.

The optimization process curves of 11 benchmark functions are shown in Fig. 3.

As can be seen from Fig. 3, for the single-peak functions, the convergence curves of the WMSDE algorithm show a monotonic downward trend, and the WMSDE algorithm can quickly reach the optimal value or continuously converge to the optimal value. For the multi-peak functions, the convergence curves of the WMSDE algorithm have many inflection points, constantly jumps out the local optimal value and approaches the global optimal value. Therefore, the experiment results show that the WMSDE algorithm has good adaptability, convergence and diversity in solving different complex function, and can effectively enhance global search ability.

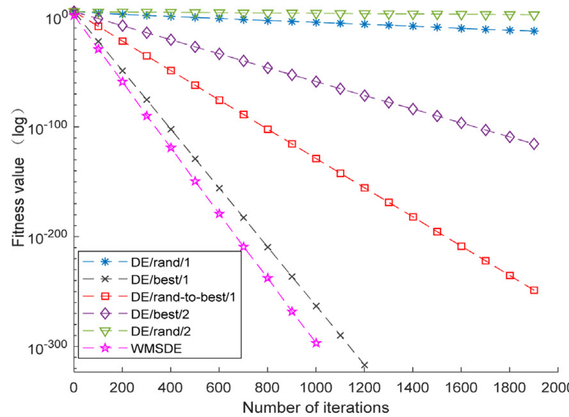
4.2.2. Result comparison and analysis

In order to further verify the optimization performance of the WMSDE algorithm, the existed DE2/F, MEDE, pADE and RMDE algorithms are selected in here [44–49]. The comparison results for the functions $f_1, f_2, f_3, f_5, f_9, f_{10}$ and f_{11} are shown in Table 4.

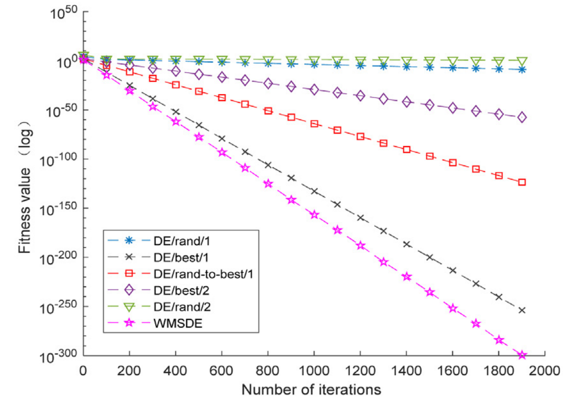
As can be seen from Table 4, for the functions f_1 and f_2, the five algorithms can obtain better optimal value, mean and standard deviation in both 30 and 50 dimensions, but the WMSDE algorithm can obtain the best optimal value(zero) in all evaluating indicators. For the function f_3, although the WMSDE algorithm cannot obtain the optimal value, but the obtained optimal value, mean and standard deviation are better than those of the four other comparison algorithms, and the standard deviation is zero in the 30 dimensions. For the function f_5, because this function is a unimodal function when the independent variable is 2∼3 dimensions, the obtained results of all algorithms are not good. When the dimension of the independent variable increases, the number of optimal values also increases. At the same time, its global optimal value is lied in a parabolic valley, which is easy to be obtained. However, it is difficult to obtain the global optimal value because of the small change of the function value in the valley. For the functions f_9, f_{10} and f_{11}, the WMSDE algorithm and pADE algorithm can obtain the best optimal value(zero) in all evaluating indicators. The two algorithms have a better optimization performance of the mean, optimal value and standard deviation than DE2/F, MEDE and RMDE algorithms in both 30 and 50 dimensions. Therefore, from the experiment results of the multi-peak functions, the WMSDE algorithm takes on better optimization performance, and can also avoid to fall into local optimum and obtain the better optimal solution for these functions. The WMSDE algorithm shows better optimization performance, stability, convergence and diversity in solving these benchmark functions.

Table 3
The experimental results of 11 benchmark functions.

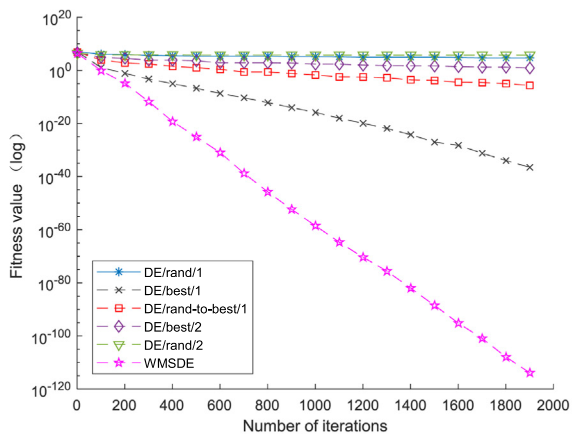
Functions	Algorithms	30 Dimensions			50 Dimensions		
		Best	Mean	Std	Best	Mean	Std
f_1	DE/rand/1	6.35E−032	1.32E−030	2.39E−030	2.72E−018	4.06E−017	4.59E−017
	DE/best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand-to-best/1	2.05E−275	2.78E−274	**0.00E−000**	4.85E−265	6.13E−264	**0.00E−000**
	DE/best/2	3.66E−143	3.94E−142	3.61E−142	6.80E−126	7.98E−125	9.53E−125
	DE/rand2	4.12E−005	1.08E−004	6.18E−05	8.07E−002	1.41E−001	4.46E−002
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_2	DE/rand/1	3.23E−012	4.34E−001	4.39E−011	2.38E−010	8.75E−010	4.86E−010
	DE/best/1	1.59E−272	2.36E−271	0.00E−000	1.86E−268	8.14E−268	**0.00E−000**
	DE/rand-to-best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand/2	1.09E−070	3.68E−070	2.59E−070	1.55E−062	8.84E−062	1.16E−061
	DE/best/2	3.80E−002	8.30E−002	4.20E−002	2.14E−000	2.84E−000	5.85E−001
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_3	DE/rand/1	5.55E+002	1.10E+003	4.89E+002	3.18E+004	5.24E+004	1.75E+004
	DE/best/1	**0.00E−000**	1.83E−083	3.42E−083	1.54E−053	7.38E−040	1.98E−039
	DE/rand-to-best/1	1.84E−031	2.55E−029	3.89E−029	1.98E−009	3.53E−008	4.80E−008
	DE/rand/2	4.55E−016	7.71E−015	1.18E−014	6.70E−002	9.90E−002	3.82E−002
	DE/best/2	1.59E+004	2.43E+004	6.03E+003	393 011.5	518 464.4	4.82E−001
	WMSDE	8.25E−173	**5.76E−142**	**0.00E−000**	**8.68E−134**	**1.20E−094**	**5.09E−094**
f_4	DE/rand/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand-to-best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_5	DE/rand/1	**6.80**E−001	4.74E+001	6.66E+001	**4.33E+001**	5.66E+001	2.00E+001
	DE/best/1	2.89E+001	2.89E+001	**1.98E−002**	4.89E+001	4.89E+001	**0.00E−000**
	DE/rand-to-best/1	2.88E+001	2.89E+001	3.65E−002	4.88E+001	4.89E+001	3.27E−002
	DE/rand/2	2.87E+001	2.88E+001	7.05E−002	4.88E+001	4.89E+001	3.39E−002
	DE/best/2	**2.48E+001**	**2.57E+001**	8.37E−001	1.30E+002	1.63E+002	3.60E+001
	WMSDE	2.86E+001	2.88E+001	1.00E−001	4.97E+001	**4.88E+001**	6.00E−002
f_6	DE/rand/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand-to-best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_7	DE/rand/1	3.35E−002	4.00E−002	5.30E−003	5.20E−002	7.00E−002	8.40E−003
	DE/best/1	6.62E−007	9.67E−006	8.24E−006	8.69E−007	7.85E−006	4.74E−006
	DE/rand-to-best/1	5.47E−007	1.45E−005	1.90E−005	4.51E−007	9.69E−006	1.08E−005
	DE/rand/2	2.66E−006	1.45E−005	8.37E−006	8.88E−007	1.06E−005	1.29E−005
	DE/best/2	5.90E−002	8.80E−002	2.84E−002	7.50E−001	1.07E−000	4.82E−001
	WMSDE	**3.27E−007**	**7.73E−006**	**7.03E−006**	**3.45E−007**	**5.56E−006**	**1.29E−006**
f_8	DE/rand/1	−1.10E+004	−7.64E+003	1.55E+003	−1.20E+004	−9.08E+003	1.64E+003
	DE/best/1	−3.30E+003	−2.92E+003	3.16E+002	−4.05E+03	−3.51E+003	3.34E+003
	DE/rand-to-best/1	−6.39E+003	−4.36E+003	8.30E+002	−5.90E+003	−5.22E+003	4.43E+002
	DE/rand/2	−4.93E+003	−4.34E+003	3.56E+002	−5.67E+003	−5.34E+003	2.55E+002
	DE/best/2	−5.18E+003	−4.89E+003	2.19E+002	−6.87E+003	−6.38E+003	3.53E+002
	WMSDE	**−1.26E+004**	**−9.70E+003**	**1.40E+002**	**−1.26E+004**	**−8.77E+003**	**2.02E+002**
f_9	DE/rand/1	2.53E+001	7.29E+001	3.10E+001	2.71E+001	1.75E+002	9.12E+001
	DE/best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand-to-best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/2	2.14E+002	2.28E+002	9.24E−000	5.41E+002	5.55E+002	2.46E+001
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_{10}	DE/rand/1	7.99E−015	7.99E−015	0.00E−000	1.10E−009	3.36E−009	2.27E−009
	DE/best/1	8.88E−016	3.32E−015	1.64E−015	8.88E−016	3.47E−015	1.51E−015
	DE/rand-to-best/1	4.44E−015	4.44E−015	**0.00E−000**	4.44E−015	4.44E−015	**0.00E−000**
	DE/rand/2	4.44E−015	4.44E−015	**0.00E−000**	4.44E−015	4.44E−015	**0.00E−000**
	DE/best/2	8.20E−003	1.18E−002	2.20E−003	4.90E−001	7.60E−001	2.06E−001
	WMSDE	**8.88E−016**	**8.88E−016**	**0.00E−000**	**8.88E−016**	**8.88E−016**	**0.00E−000**
f_{11}	DE/rand/1	1.28E−010	5.31E−003	4.07E−003	**0.00E−000**	4.44E−017	7.36E−017
	DE/best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand-to-best/1	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/rand/2	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	DE/best/2	3.73E−006	2.81E−005	3.46E−005	2.15E−003	1.32E−002	3.79E−003
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**



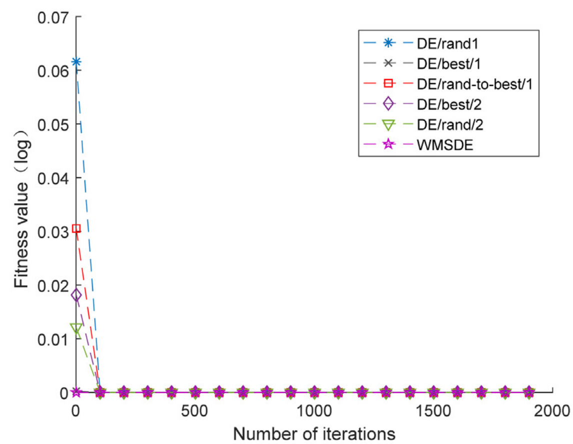
(a) Optimization process curve of f_1



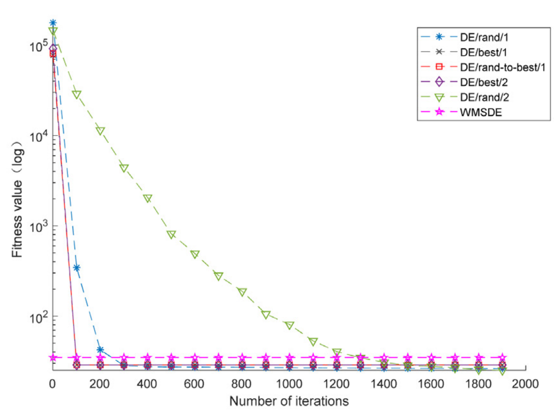
(b) Optimization process curve of f_2



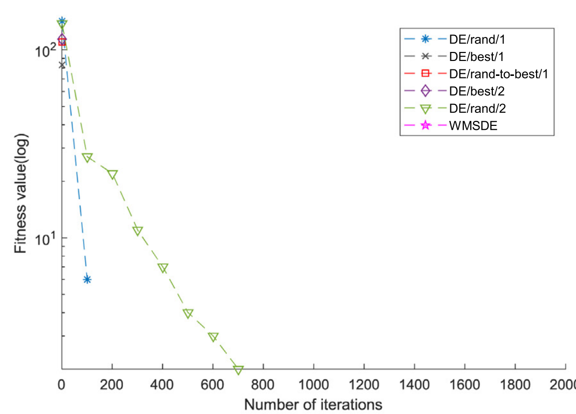
(c) Optimization process curve of f_3



(d) Optimization process curve of f_4



(e) Optimization process curve of f_5



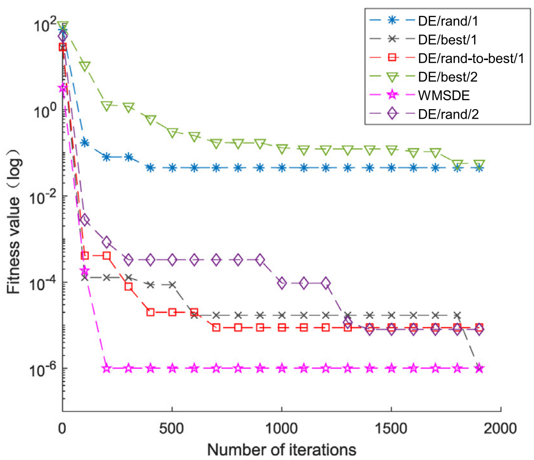
(f) Optimization process curve of f_6

Fig. 3. Optimization process curves of 11 functions.

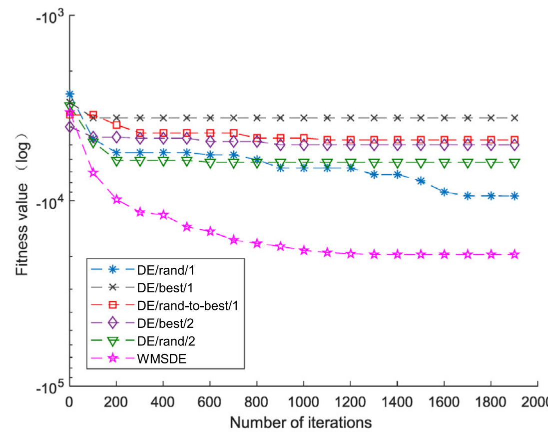
5. Airport gate assignment method

5.1. Construct mathematical modeling of gate assignment

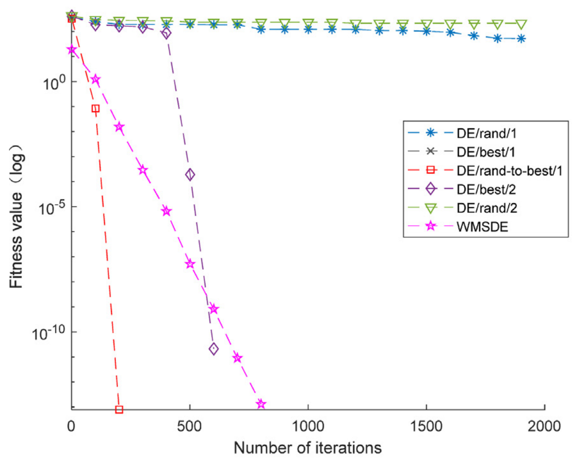
As an important resource of the airport, the gate is the key factor to realize the fast and safe flight parking. Airport gate assignment refers to assign a specific gate for each flight according to the attributes of flight and aircraft type. It should not only provide better service for passengers and save costs for airlines from the perspective of passengers and airlines, but also reasonably, evenly and efficiently assign limited gate resources from the perspective of airport operation control department, so as to prevent adverse effect of emergencies on airport operation. In the airport gate assignment, due to comprehensively considering the interests of many companies, the airport gate assignment is a multi-objective optimization problem. The satisfaction degree of passengers to airport service is very important for the airport operation, and the walking distance is directly related to the evaluation of the airport by passengers. Therefore, the



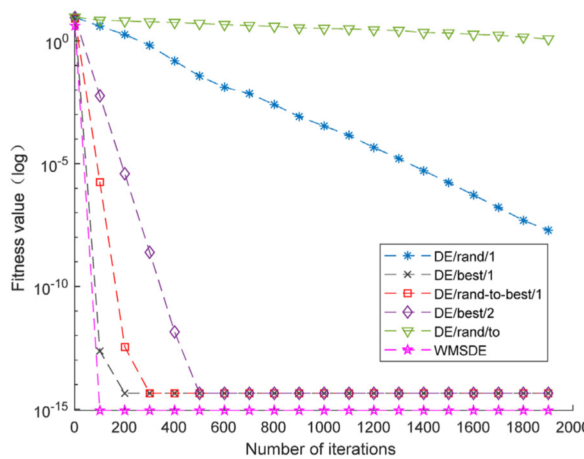
(g) Optimization process curve of f_7



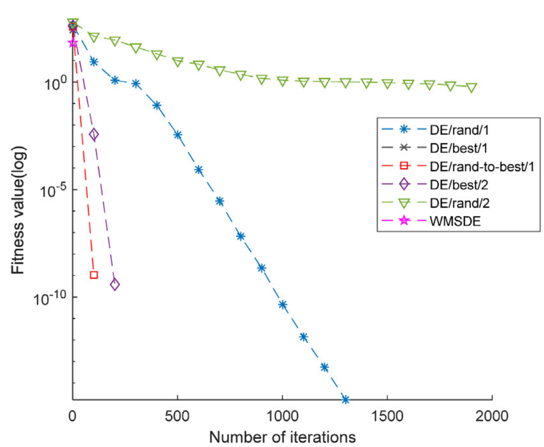
(h) Optimization process curve of f_8



(i) Optimization process curve of f_9



(j) Optimization process curve of f_{10}



(k) Optimization process curve of f_{11}

Fig. 3. (*continued*).

shortest total walking distance for passengers is selected as the optimization objective function. When the flight encounters some small-scale short-term delays, it is hoped that a little adjustment can ensure the normal operation of the flight and the balanced utilization of all gates. Therefore, the most balanced idle time for each gate is selected as the optimization objective function. Large gates can be assigned to all aircrafts, while small gates can only be assigned to small aircrafts. In actual operation, if large gates are occupied too early by small and medium-sized aircrafts, then the distribution options of later large aircrafts will be less, and even the large aircrafts are forced to be assigned to the apron. At the same time, large flights often take more passengers, once they are assigned to the apron, it will cause more inconvenience to passengers and satisfaction. Therefore, the best use of large gates is selected as the optimization objective function. From the perspective of airport management,

Table 4
The comparison results among different algorithms.

Functions	Algorithm	30 dimensions			50 dimensions		
		Best	Mean	Std	Best	Mean	Std
f_1	DE2/F	2.19E−039	1.50E−038	1.32E−039	2.45E−028	7.17E−028	6.68E−015
	MEDE	1.14E−065	3.21E−064	2.93E−064	2.18E−041	1.43E−040	5.19E−020
	pADE	4.15E−245	2.41E−253	**0.00**E−000	3.55E−249	1.21E−246	**0.00E−000**
	RMDE	5.32E−159	3.16E−144	2.40E−143	2.57E−076	1.12E−063	1.88E−011
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_2	DE2/F	3.28E−024	2.01E−023	4.36E−024	8.75E−018	1.32E−017	3.60E−018
	MEDE	9.37E−034	3.08E−033	1.93E−033	1.59E−022	4.41E−022	1.84E−022
	pADE	3.21E−115	1.56E−113	3.03E−113	1.70E−109	2.85E−108	2.98E−108
	RMDE	2.44E−054	5.01E−044	1.42E−043	3.38E−022	8.31E−018	3.16E−017
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_3	DE2/F	3.02E−001	4.22E−001	7.98E−002	1.88E+002	2.82E+002	4.28E+001
	MEDE	1.02E−000	4.44E−000	2.64E−000	1.08E+002	1.92E+002	4.32E+001
	pADE	5.72E−048	5.02E−042	1.80E−041	8.85E−044	8.85E−044	5.10E−036
	RMDE	2.84E−034	1.58E−029	4.51E−029	4.56E−014	4.56E−014	5.34E−008
	WMSDE	**8.25E−173**	**5.76E−142**	**0.00E−000**	**8.68E−134**	**1.20E−094**	**5.09E−094**
f_5	DE2/F	2.45E+001	5.39E+001	2.30E+001	3.49E+001	7.92E+001	2.72E+001
	MEDE	2.09E−001	2.18E+001	1.84E+001	1.78E+001	4.56E+001	1.97E+001
	pADE	2.83E+001	2.85E+001	8.91E−002	4.82E+001	4.84E+001	9.50E+001
	RMDE	**6.21E−013**	**4.72E−005**	**2.06E−004**	**6.60E−007**	**9.33E−002**	**3.02E−001**
	WMSDE	2.86E+001	2.88E+001	1.00E−001	4.97E+001	4.88E+001	6.00E−002
f_9	DE2/F	2.71E−027	7.70E−025	1.82E−024	**0.00E−000**	8.97E−001	8.48E−001
	MEDE	1.11E−038	9.16E−017	2.99E−016	1.56E+002	9.19E+000	9.19E+000
	pADE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	RMDE	1.78E−015	4.88E−013	1.93E−012	2.49E−014	4.83E−012	4.83E−012
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
f_{10}	DE2/F	8.01E−015	8.12E−015	**0.00E−000**	2.23E−014	2.31E−014	2.26E−015
	MEDE	4.44E−015	7.46E−015	1.30E−015	7.99E−015	1.17E−014	2.93E−015
	pADE	**8.88E−016**	**8.88E−016**	**0.00E−000**	**8.88E−016**	**8.88E−016**	**0.00E−000**
	RMDE	2.22E−014	1.09E−010	4.05E−010	9.33E−014	1.23E−011	3.09E−011
	WMSDE	**8.88E−016**	**8.88E−016**	**0.00E−000**	**8.88E−016**	**8.88E−016**	**0.00E−000**
f_{11}	DE2/F	**0.00E−000**	1.03E−007	3.32E−007	**0.00E−000**	6.79E−007	2.06E−006
	MEDE	**0.00E−000**	1.11E−003	2.71E−003	**0.00E−000**	8.63E−004	2.68E−003
	pADE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**
	RMDE	1.11E−016	1.54E−015	1.50E−015	1.67E−015	1.48E−003	4.62E−003
	WMSDE	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**	**0.00E−000**

to meet the safety of flight operation, the same types of flights are assigned to the corresponding gates to greatly improve the utilization efficiency of gates and save the operating cost of the gate. It can also help the airport to assign more flights and expand the number of flights for airport. Therefore, the highest occupancy efficiency based on gates is selected as the optimization objective function. In the process of gate assignment, the gates should be assigned to the corresponding gates as far as possible. Therefore, the minimum matching difference between flights and gates is selected as the optimization objective function. All in all, the gate assignment problem need to comprehensively consider the interests of passengers, airport, airlines and government. Therefore, it is a complex multi-objective optimization problem. Each optimization objective function is constructed as follows.

5.1.1. The objective function of the most balanced idle time for gates

The objective function of the most balanced idle time of gates is described as follow.

$$F_1 = min \sum_{i=1}^{n} \sum_{j=1}^{m} S_{ij}^2 + \sum_{j=1}^{m} SS_j^2 \tag{11}$$

where n denotes the total number of flights, m denotes the number of gates. S_{ij} denotes the idle time of gate when the flight i arrives at the gate j. SS_j denotes the idle time of the gate.

5.1.2. The objective function of the shortest total walking distance for passengers

The objective function of the shortest walking distance for passengers is described as follow.

$$F_2 = min \sum_{i=1}^{n} \sum_{j=1}^{m} q_{ij}\, f_j\, y_{ij} \tag{12}$$

where q_{ij} is the number of passengers of flight i at gate j. f_j is the walking distance of passengers from security checkpoint to gate j. y_{ij} is a variable of 0–1.

5.1.3. The objective function of the best use of large gates

The objective function of the best use of large gates is described as follow.

$$F_3 = \sum_{i=1}^{n} \sum_{j=1}^{m} G_{ij} \tag{13}$$

where G_{ij} are the parked small and medium-sized aircraft in large gate and the parked small aircrafts in medium gate.

5.1.4. The objective function of the highest occupancy efficiency based on gates

The objective function of the highest occupancy efficiency based on gates is described as follow.

$$F_4 = -\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{(t^{a}{}_{ij} - t_{ij}^{b}) \times y_{ij}}{T} \tag{14}$$

5.1.5. The objective function of the minimum matching difference between flights and gates

The objective function of the minimum matching difference between flights and gates is described as follow.

$$F_5 = \sum_{i=1}^{n}\sum_{j=1}^{m} \rho_{ij} y_{ij} \tag{15}$$

5.1.6. Constraints

The constraints are to limit the calculation process, so that the calculation result can meet the requirements. In the same airport, the constraints of different gates and flights may be different. The constraints include gate constraints, flight constraints and so on in this paper.

(1) Each flight must only be assigned to one gate.

$$\sum_{j}^{N} x_{ij} = 1, \quad \forall i \in F, j \in G \tag{16}$$

If the flight i is assigned to gate j, then $x_{ij} = 1$. Otherwise there is $x_{ij} = 0$.

(2) The gate constraints to aircraft type

When the flight i is assigned to gate j, it must comply with $\varepsilon_i \leq \rho_j + (1 - y_{ij})\Omega$. Where ε_i is aircraft type of flight i, ρ_j is the gate j for allowing the largest aircraft type, Ω is arbitrarily positive number.

(3) $|A_i - D_i| \geq T$

The interval time between the two adjacent flights at the same gate must be greater than the safety interval time. T is a safety interval time at same gate.

(4) 0–1 variable constraints x_{ij}, p_{ij}, G_{ij}, $y_{ij} \in \{0, 1\}$.

(5) Positive integer constraints: S_{ij}, t_{ij}, SS_j, $f_j \geq 0$.

5.1.7. Non-quantization processing of objective function

Because the objective functions $[F_1(x), F_2(x), \ldots, F_n(x)]$ have their own targets, units and dimensions, it is necessary to deal with these objective functions quantitatively. In dealing with the multi-objective optimization problems, the commonly used methods are linear weighting method, square sum weighting method and constraint method and so on. Since the linear weighting method has the characteristics of convenient operation and good effect, the linear weighting method is selected to deal with the multi-objective optimization model. The weight factor is set as $W_i \geq 0 (i = 1, 2, \ldots, n)$.

Set $F_i^0 = \max[|F_i|]$ and $F_i^0 \neq 0$. The objective function of non-quantization is described as follow.

$$F = \sum_{i=1}^{n} \frac{W_i F_i}{F_i^0} \tag{17}$$

Therefore, the mathematical modeling of gate assignment problem is described as follow.

$$\begin{aligned} F = {} & \frac{W_1}{F_1^0}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij}^2 + \sum_{j=1}^{m} SS_j^2\right] + \frac{W_2}{F_2^0}\sum_{i=1}^{n}\sum_{j=1}^{m} q_{ij}\, f_j\, y_{ij} \\ & - \frac{W_3}{F_3^0}\sum_{i=1}^{n}\sum_{j=1}^{m} G_{ij} - \frac{W_4}{F_4^0}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{(t_{ij}^{a} - t_{ij}^{ab}) \times y_{ij}}{T} \\ & + \frac{W_5}{F_5^0}\sum_{i=1}^{n}\sum_{j=1}^{m} \rho_{ij}\, y_{ij} \end{aligned} \tag{18}$$

5.2. Airport gate assignment method using WMSDE algorithm

5.2.1. Airport gate assignment method

Gate assignment problem is a NP-hard problem with complex constraints and large scale. It is difficult to find the accurate optimal solution by using traditional methods. The general intelligent optimization algorithm is also difficult to find the accurate optimal solution. The proposed WMSDE algorithm has the characteristics of less undetermined parameters, fast convergence speed and good robustness. Therefore, the proposed WMSDE algorithm is applied to solve the multi-objective optimization model of airport gate assignment. A fast assignment method of airport gate assignment based on WMSDE algorithm is proposed, which can effectively realize the assignment of airport gate assignment and obtain the optimal airport gate assignment scheme.

5.2.2. Airport gate assignment model

The flow of airport gate assignment method by using the proposed WMSDE algorithm is shown Fig. 4.

5.2.3. Steps of airport gate assignment

The detailed steps of airport gate assignment method by using the proposed WMSDE are described as follows.

Step 1. Initialize the parameters of the WMSDE algorithm, including population size, scaling factor, mutation strategy, crossover probability, maximum number of iterations and other parameters.

Step 2. The initial population is divided into several subpopulations according to the solving complex optimization problem.

Step 3. The subpopulations are evaluated to construct binary matrix. The binary string of each row is converted into a decimal number, which is selected as the current gates to obtain the initial gate matrix.

Step 4. Read the information of gates and flights, the conflicting flights are eliminated and the gate matrix is adjusted according to the constraints.

Step 5. The objective function of airport gate assignment problem is solved to obtain the minimum value of the objective function by using the WMSDE algorithm.

Step 6. Record and save the optimal value and the assigned gate matrix.

Step 7. The conflicting flights are eliminated to obtain a new gate assignment scheme.

Step 8. Determine whether the maximum number of iterations is reached. When the end condition is met, the optimal gate assignment result is output. Otherwise, go to **Step 4**.

6. Application case and analysis

6.1. Experimental data and environment

In order to validate the effectiveness of the proposed gate assignment method in solving actual gate assignment engineering problem, an airport gate assignment case with 250 flights and 30 gates from Guangzhou Baiyun Airport on July 26, 2015 is used to test and simulate systematically. The detailed information of 30 gates are described in Table 5, and the detailed information of 250 flights are described in Table 6. The safe interval time between two adjacent flights is 5 min for the same gate. The gates and flights can be divided into large, medium and small gates and flights. The large gates can spark all flights, the medium gates can spark the medium flights and small flights, and the small gates only spark the small flights. When the flights are not assigned to the gates, they will be assigned to the apron.

The parameters of the WMSDE algorithm are set as follows. The population size is 250, individual dimension is 5, and the

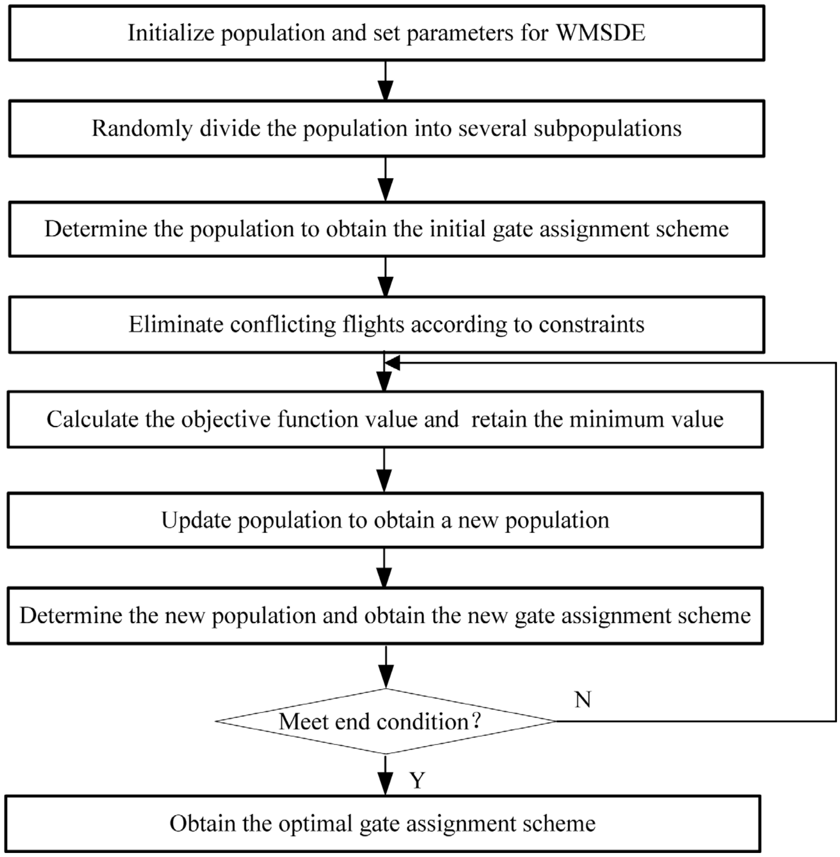


Fig. 4. The flow of airport gate assignment using WMSDE.

Table 5
The detailed information of gates.

Gate	Walking distance (m)	Gate type	Gate	Walking distance (m)	Gate type
1	190	M	16	115	L
2	975	M	17	215	M
3	400	L	18	535	S
4	333	M	19	1050	M
5	260	L	20	170	M
6	135	S	21	585	L
7	1100	M	22	1250	M
8	150	M	23	500	L
9	384	L	24	920	L
10	960	M	25	270	L
11	1000	S	26	230	M
12	235	L	27	265	L
13	1200	S	28	450	L
14	580	L	29	1300	M
15	440	L	30	426	L

Table 6
The detailed information of flights.

Flights	Arrived time	Left time	Passengers	Flight type
1	2015-7-26 0:05:00	2015-7-26 5:15:00	482	L
2	2015-7-26 0:05:00	2015-7-26 5:45:00	273	M
3	2015-7-26 0:10:00	2015-7-26 5:30:00	261	M
4	2015-7-26 0:15:00	2015-7-26 5:30:00	116	M
5	2015-7-26 0:15:00	2015-7-26 5:15:00	244	M
6	2015-7-26 0:20:00	2015-7-26 5:30:00	312	L
7	2015-7-26 0:25:00	2015-7-26 5:20:00	340	L
8	2015-7-26 0:30:00	2015-7-26 6:00:00	198	M
⋮	⋮	⋮	⋮	⋮
249	2015-7-26 23:50:00	2015-07-27 01:50:00	252	S
250	2015-7-26 23:55:00	2015-7-27 9:10:00	378	M

maximum number of iterations is 200. The experiment is executed 20 times. The experimental environments are Intel (R) core (TM) i5-7400 CPU 3.00 GHz, 8G RAM, Windows 10, and MATLAB R2018a.

6.2. Assigned results of gates

The proposed WMSDE algorithm is used to solve the multi-objective optimization model of airport gate assignment. The algorithm has been independently implemented 20 times to obtain 20 experiment results. Then best one of 20 experiment results is selected to analyze. The obtained gate assignment result is shown in Table 7., and the corresponding Gantt chart is shown in Fig. 5.

In order to intuitively observe the occupancies of flights in each gate, the number of assigned flights for each gate is shown in Fig. 6.

The process curve of optimal value in solving multi-objective optimization model of airport assignment using WMSDE algorithm is shown Fig. 7.

As can be seen from Table 7 and Figs. 5–7, 244 flights are assigned to 30 gates, 6 flights are assigned to the apron. The airport gate assignment rate reaches at 97.6%. From the number of assigned flights for each gate, the number of assigned flights for each gate is more balanced, and the idle time of each gate is more balanced, which enables staff to have sufficient time to schedule, and avoids the idle waste of gate and does not overuse, and makes the airline resources to obtain a more reasonable assignment and utilization. At the same time, the proposed WMSDE algorithm obtains the optimal objective value (0.6256) at the 83th iteration. Therefore, the proposed WMSDE algorithm is used

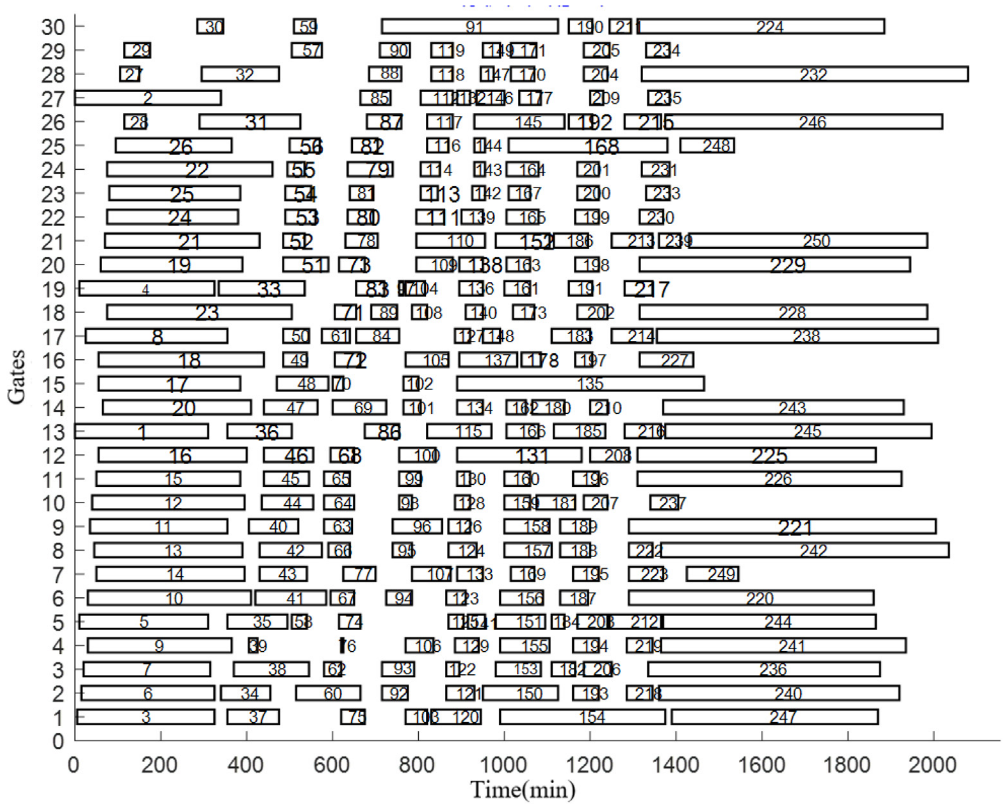


Fig. 5. The corresponding Gantt chart.

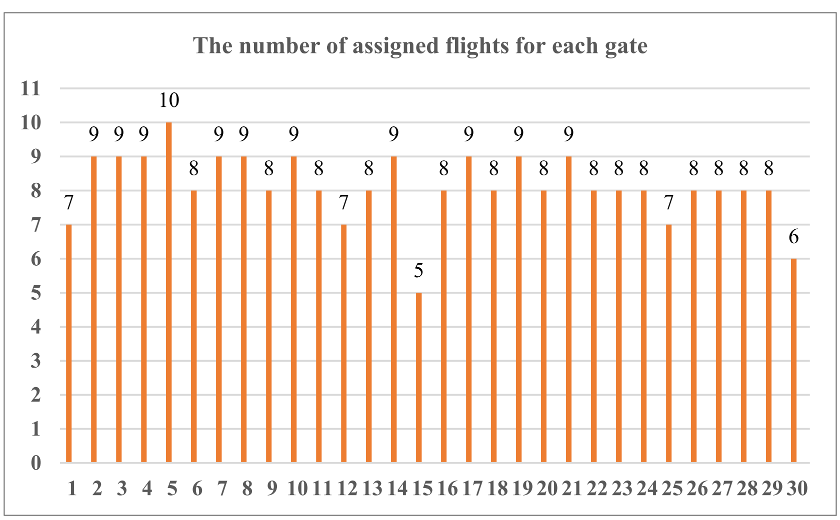


Fig. 6. The number of assigned flights for each gate.

to solve the multi-objective optimization model of airport gate assignment, which can obtain better gate assignment results. It takes on better optimization performance, convergence behavior and diversity. Therefore, the proposed WMSDE algorithm can effectively solve real-world practical engineering problems, and obtain better optimization solution.

6.3. Comparison and analysis

In order to further demonstrate the optimization performance of the proposed WMSDE algorithm, the DE, DE2/F, MEDE, pADE and RMDE algorithms are selected to solve the multi-objective optimization model of airport gate assignment. The comparison results of ten times by using six algorithms are shown in Table 8, Figs. 8 and 9.

As can be seen from Table 8, Figs. 8 and 9, the proposed WMSDE algorithm is used to solve the constructed multi -objective optimization model of gate assignment, the minimum optimal value, average optimal value, maximum assignment rate and average assignment rate are 0.6256, 0.6302, 97.6% and 96.54%, respectively. For the DE, DE2/F, MEDE, pADE, RMDE and WMSDE, the minimum optimal objective value, average optimal objective value, maximum gate assignment rate and average gate assignment rate of the WMSDE algorithm is better than those of the DE, DE2/F, MEDE, pADE and RMDE. That is to say, the WMSDE algorithm obtains the best quality of the solution.

Table 7
The obtained gate assignment results.

Gate	Assigned flights										Total
1	3	37	75	103	120	154	247				7
2	6	34	60	92	121	150	193	218	240		9
3	7	38	62	93	122	153	182	206	236		9
4	9	39	76	106	129	155	194	219	241		9
5	5	35	58	74	125	141	151	184	203	212	10
6	10	41	67	94	123	156	187	220			8
7	14	43	77	107	133	169	195	223	249		9
8	13	42	66	95	124	157	188	222	242		9
9	11	40	63	96	126	158	189	221			8
10	12	44	64	98	128	159	181	207	237		9
11	15	45	65	99	130	160	196	226			8
12	16	46	68	100	131	208	225				7
13	1	36	86	115	166	185	216	245			8
14	20	47	69	101	134	162	180	210	243		9
15	17	48	70	102	135						5
16	18	49	72	105	137	178	197	227			8
17	8	50	61	84	127	148	183	214	238		9
18	23	71	89	108	140	173	202	228			8
19	4	33	83	97	104	136	161	191	217		9
20	19	51	73	109	138	163	198	229			8
21	21	52	78	110	152	186	213	239	250		9
22	24	53	80	111	139	165	199	230			8
23	25	54	81	113	142	167	200	233			8
24	22	55	79	114	143	164	201	231			8
25	26	56	82	116	144	168	248				7
26	28	31	87	117	145	192	215	246			8
27	2	85	112	132	146	177	209	235			8
28	27	32	88	118	147	170	204	232			8
29	29	57	90	119	149	171	205	234			8
30	30	59	91	190	211	224					6
Total											244

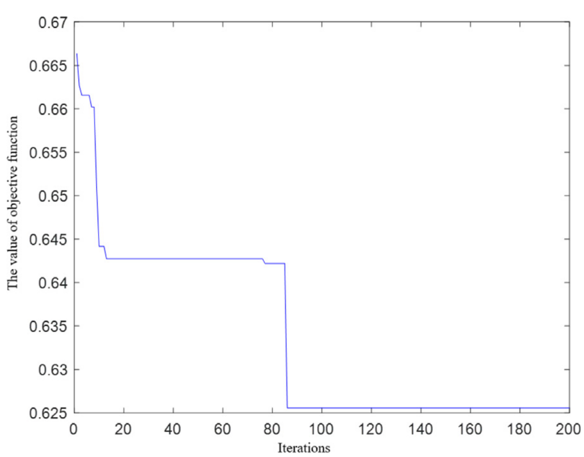


Fig. 7. The process curve of optimal value.

Table 8
The calculation and comparison results.

Algorithm	Minimum value	Average value	Average run time (s)	Maximum assignment rate (%)	Average assignment rate (%)
DE	0.7453	0.7857	14.427	89.6	88.66
DE2/F	0.7181	0.7362	14.943	92.4	91.88
MEDE	0.6535	0.6816	17.481	95.6	95.24
pADE	0.6429	0.6579	15.673	94.4	93.88
RMDE	0.6392	0.6483	16.730	96.8	96.04
WMSDE	0.6256	0.6302	16.124	97.6	96.54

But from the experiment results, we can also see that average run time of the WMSDE is worse than those of the DE, DE2/F and pADE algorithms. In general, although the WMSDE needs more time to solve the constructed multi-objective optimization

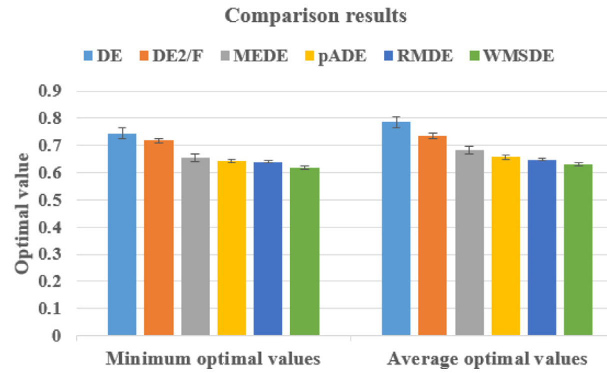


Fig. 8. The comparison results of the minimum optimal values and average optimal values by using six algorithms.

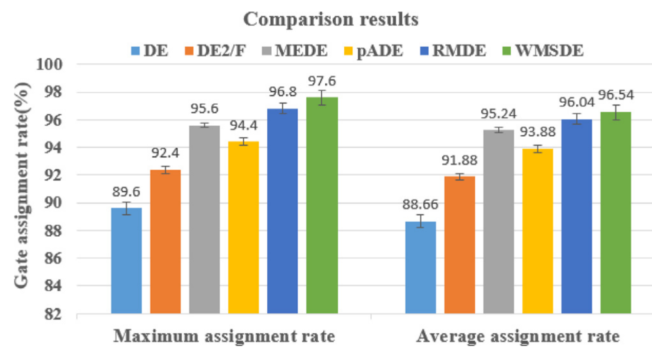


Fig. 9. The comparison results of the maximum assignment rate and average assignment rate by using six algorithms.

model of gate assignment, the solution quality of the WMSDE has been improved by comparing the solution quality of the other algorithms. The WMSDE can reduce the costs of the airport and airline, and improve the comprehensive service level. Therefore, the constructed multi-objective optimization model of gate assignment and the proposed WMSDE can significantly reduce the walking distances for passengers, balance the idle time for gates, make use of large gates, improve occupancy efficiency based on gates and best match between flights and gates. At the same time, the service level of airport and satisfaction degree of passengers are improved. The proposed gate assignment method can effectively improve the flexibility for gate assignment and avoid to occur a large number of flight delays. It effectively provides a valuable reference for assigning the gates.

7. Conclusion and future work

For the existing problems of solution accuracy and poor convergence of the DE algorithm, a new optimal mutation strategy based on the complementary advantages of five mutation strategies in the first generation is designed. Then a novel improved differential evolution (WMSDE) algorithm based on the wavelet basis function and new optimal mutation strategy is proposed to improve the search quality, accelerate convergence and avoid fall into local optimum and stagnation. 11 benchmark functions and an actual airport gate assignment problem are used to validate the effectiveness of the proposed WMSDE algorithm. From the experiment results, the proposed WMSDE algorithm has a monotonous decreasing trend of convergence curve for the single-peak functions, and can quickly obtain the optimal value or converge to the optimal value. For the multimodal functions, the proposed WMSDE algorithm has multiple inflection points in the convergence curve, and can continuously jump out from the local optimal value, and approach to the global optimization solution. For airport gate assignment engineering problem, the gate assignment rate reaches at 97.6%, it significantly reduces the walking distance for passengers, balances the idle time for gates, makes use of large gates, improves occupancy efficiency based on gates and best matches between flights and gates. Therefore, the WMSDE algorithm can effectively avoid premature convergence, balance local search ability and global search ability, and also effectively solve the airport gate assignment problem and obtain ideal airport gate assignment results. This study provides a good choice for solving large-scale complex optimization problems.

Because the proposed WMSDE algorithm in solving benchmark functions and an actual airport gate assignment problem needs more running time to obtain the optimization results, it is important to how to reduce the time complexity of the proposed WMSDE algorithm in the future works.

CRediT authorship contribution statement

Wu Deng: Conceptualization, Methodology, Funding acquisition, Writing - original draft. **Junjie Xu:** Data curation, Visualization, Investigation. **Yingjie Song:** Software, Validation, Formal analysis. **Huimin Zhao:** Writing - review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61771087, under Grant 51879027; the Open Project Program of State Key Laboratory of Mechanical Transmissions of Chongqing University, China under Grant SKLMT-KFKT-201803, and Traction Power State Key Laboratory of Southwest Jiaotong University, China under Grant TPL2002, the Open Subject Project of The Key Laboratory of Air Traffic Control Operation Planning and Safety Technology of CAUC under Grant 600001010932.

References

[1] W. Deng, H.M. Zhao, L. Zou, G.Y. Li, X.H. Yang, D.Q. Wu, A novel collaborative optimization algorithm in solving complex optimization problems, Soft Comput. 21 (15) (2017) 4387–4398.

[2] M.A. Al-Betar, I. Aljarah, M.A. Awadallah, H. Faris, S. Mirjalili, Adaptive β-hill climbing for optimization, Soft Comput. 23 (24) (2019) 13489–13512.

[3] W. Deng, J. Xu, Y. Song, et al., An effective improved co-evolution ant colony optimization algorithm with multi-strategies and its application, Int. J. Bio-Inspired Comput. 2020 (2020) 1–10.

[4] Y. Liu, Y. Mu, K. Chen, et al., Daily activity feature selection in smart homes based on pearson correlation coefficient, Neural Process. Lett. 51 (2020) 1771–1787.

[5] H. Chen, Q. Zhang, J. Luo, et al., An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine, Appl. Soft Comput. 86 (2020) 1–24.

[6] H. Sharifipour, M. Shakeri, H. Haghighi, Structural test data generation using a memetic ant colony optimization based on evolution strategies, Swarm Evol. Comput. 40 (2018) 76–91.

[7] Y. Xu, H. Chen, A.A. Heidari, J. Luo, Q. Zhang, X. Zhao, C. Li, An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks, Expert Syst. Appl. 129 (2019) 135–155.

[8] M.A. Awadallah, M.A. Al-Betar, A.L.A. Bolaji, I.A. Doush, A.I. Hammouri, M. Mafarja, Island artificial bee colony for global optimization, Soft Comput. (2020) http://dx.doi.org/10.1007/s00500-020-04760-8.

[9] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, X. Zhang, Enhanced Moth-flame optimizer with mutation strategy for global optimization, Inform. Sci. 492 (2019) 181–203.

[10] M.A. Al-Betar, M.A. Awadallah, I.A. Doush, A.I. Hammouri, M. Mafarja, Z.A.A. Alyasseri, Island flower pollination algorithm for global optimization, J. Supercomput. 75 (8) (2019) 5280–5323.

[11] T. Li, J. Shi, X. Li, et al., Image encryption based on pixel-level diffusion with dynamic filtering and DNA-level permutation with 3D Latin cubes, Entropy 21 (2019) 1–21.

[12] W. Deng, J. Xu, H. Zhao, An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem, IEEE Access 7 (2019) 20281–20292.

[13] R. Chen, S. Guo, X.Z. Wang, et al., Fusion of multi-RSMOTE with fuzzy integral to classify bug reports with an imbalanced distribution, IEEE Trans. Fuzzy Syst. 27 (2019) 2406–2420.

[14] M.A. Al-Betar, M.A. Awadallah, Island bat algorithm for optimization, Expert Syst. Appl. 107 (2018) 126–145.

[15] T. Li, Z. Qian, T. He, Short-term load forecasting with improved CEEMDAN and GWO-based multiple kernel ELM, Complexity 2020 (2020) 1–20.

[16] M.A. Al-Betar, M.A. Awadallah, H. Faris, I. Aljarah, A.I. Hammouri, Natural selection methods for Grey Wolf Optimizer, Expert Syst. Appl. 113 (2018) 481–498.

[17] R.H. Huang, T.H. Yu, An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting, Appl. Soft Comput. 57 (2017) 642–656.

[18] Y. Liu, X. Wang, Z. Zhai, et al., Timely daily activity recognition from headmost sensor events, ISA Trans. 94 (2019) 379–390.

[19] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[20] C.T. Su, C.S. Lee, Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution, IEEE Trans. Power Deliv. 18 (3) (2003) 1022–1027.

[21] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Trans. Syst. Man Cybern. A 38 (1) (2008) 218–237.

[22] M.Y. Lai, E. Cao, An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows, Eng. Appl. Artif. Intell. 23 (2) (2010) 188–195.

[23] B. Dorronsoro, P. Bouvry, Improving classical and decentralized differential evolution with new mutation operator and population topologies, IEEE Trans. Evol. Comput. 15 (1) (2011) 67–98.

[24] S.M. Elsayed, R.A. Sarker, D.L. Essam, An improved self-adaptive differential evolution algorithm for optimization problems, IEEE Trans. Ind. Inf. 9 (1) (2013) 89–99.

[25] G.B. Jia, Y. Wang, Z.X. Cai, et al., An improved (mu+lambda)-constrained differential evolution for constrained optimization, Inform. Sci. 222 (2013) 302–322.

[26] W.Y. Gong, Z.H. Cai, Parameter optimization of PEMFC model with improved multi-strategy adaptive differential evolution, Eng. Appl. Artif. Intell. 27 (2014) 28–40.

[27] L.X. Tang, Y. Zhao, J.Y. Liu, An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production, IEEE Trans. Evol. Comput. 18 (2) (2014) 209–225.

[28] X.Y. Zhang, H.B. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft Comput. 26 (2015) 270–284.

[29] A.W. Mohamed, An improved differential evolution algorithm with triangular mutation for global numerical optimization, Comput. Ind. Eng. 85 (2015) 359–375.

[30] W.Z. Yi, Y.Z. Zhou, L. Gao, et al., An improved adaptive differential evolution algorithm for continuous optimization, Expert Syst. Appl. 44 (2016) 1–12.

[31] S.M. Guo, P.H. Hsu, C.C. Yang, et al., Constrained min–max optimization via the improved constraint-activated differential evolution with escape vectors, Expert Syst. Appl. 46 (2016) 336–345.

[32] M.N. Tian, X.B. Gao, C. Dai, Differential evolution with improved individual-based parameter setting and selection strategy, Appl. Soft Comput. 56 (2017) 286–297.

[33] Z.Q. Cai, L. Lv, H. Huang, et al., Improving sampling-based image matting with cooperative coevolution differential evolution algorithm, Soft Comput. 21 (15) (2017) 4417–4430.

[34] N.H. Awad, M.Z. Ali, R.M. Duwairi, Ulti-objective differential evolution based on normalization and improved mutation strategy, Nat. Comput. 16 (4) (2017) 661–675.

[35] V. Ho-Huu, T. Nguyen-Thoi, T. Truong-Khac, et al., An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints, Neural Comput. Appl. 29 (1) (2018) 167–185.

[36] M.S. Maucec, J. Brest, B. Boskovic, et al., Improved differential evolution for large-scale black-box optimization, IEEE Access 6 (2018) 29516–29531.

[37] S.H. Wang, Y.Z. Li, H.Y. Yang, et al., Self-adaptive differential evolution algorithm with improved mutation strategy, Soft Comput. 22 (10) (2018) 3433–3447.
[38] S. Ajithapriyadarsini, P.M. Mary, M.W. Iruthayarajan, Automatic generation control of a multi-area power system with renewable energy source under deregulated environment: adaptive fuzzy logic-based differential evolution (DE) algorithm, Soft Comput. 23 (22) (2019) 12087–12101.
[39] S. Yazdani, E. Hadavandi, LMBO-DE: a linearized monarch butterfly optimization algorithm improved with differential evolution, Soft Comput. 23 (17) (2019) 8029–8043.
[40] R. Vafashoar, M.R. Meybodi, A multi-population differential evolution algorithm based on cellular learning automata and evolutionary context information for optimization in dynamic environments, Appl. Soft Comput. J. 88 (2020) 106009.
[41] S.L. Wang, F. Morsidi, T.F. Ng, H. Budiman, S.C. Neoh, Insights into the effects of control parameters and mutation strategy on self-adaptive ensemble-based differential evolution, Inform. Sci. 514 (2020) 203–233.
[42] G.N. Ben, An accelerated differential evolution algorithm with new operators for multi-damage detection in plate-like structures, Appl. Math. Model. 80 (2020) 366–383.
[43] W. Deng, H. Liu, J. Xu, et al., An improved quantum-inspired differential evolution algorithm for deep belief network, IEEE Trans. Instrum. Meas. 2020 (2020) 1–8.
[44] A. Ghosh, S. Das, A. Chowdhury, R. Giri, An improved differential evolution algorithm with fitness-based adaptation of the control parameters, Inform. Sci. 181 (18) (2011) 3749–3765.
[45] G. Hafeez, K.S. Alimgeer, A.B. Qazi, et al., A hybrid approach for energy consumption forecasting with a new feature engineering and optimization framework in smart grid, IEEE Access 8 (2020) 96210–96226.
[46] H. Zhao, J. Zheng, W. Deng, et al., Semi-supervised broad learning system based on manifold regularization and broad network, IEEE Trans. Circuits-I 67 (2020) 983–994.
[47] Z.Y. Meng, J.S. Pan, K.K. Tseng, PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl.-Based Syst. 168 (2019) 80–99.
[48] C. Wang, Y.C. Liu, Q.J. Zhang, et al., Association rule mining based parameter adaptive strategy for differential evolution algorithms, Expert Syst. Appl. 123 (2019) 54–69.
[49] W. Deng, H.M. Zhao, X.H. Yang, J.X. Xiong, M. Sun, B. Li, Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment, Appl. Soft Comput. 59 (2017) 288–302.