

## Chapter 2

# Fireworks Algorithm (FWA)

Inspired by fireworks explosions in the sky at night, the fireworks algorithm (FWA) was proposed by the author in 2010, through the observation of the fact that fireworks explosion is similar to the way an individual searches for optimal solution in swarm intelligence algorithms. Recently, FWA has received extensive concerns from many active researchers in the swarm intelligence community. This chapter presents the fundamental principle, main constitution, implementation, and performance of the FWA, aiming to elaborate the FWA systematically and completely. The main contents include the key components, realization, characteristic, and impact of operations of FWA, as well as comparisons with genetic algorithm and particle swarm optimization.

### 2.1 Introduction

Setting off fireworks is an important creative and joyful activity during Spring Festival in China. At this time, tens of thousands of fireworks explode in the night sky and show beautiful patterns of sparks. Usually, fireworks of different prices and specifications produce entirely different patterns. For example, fireworks of lower price produce less sparks with larger amplitude compared with higher price fireworks and vice versa.

The way fireworks explode is similar to the way an individual searches the optimal solution in swarm intelligence algorithms. As a swarm intelligence algorithm, fireworks algorithm consists of four parts, i.e., the explosion operator, mutation operator, mapping rule and selection strategy. The effect of the explosion operator is to generate sparks around fireworks. The number and amplitude of the sparks are governed by the explosion operator. After that, some sparks are produced by mutation operator. The mutation operator utilizes Gaussian operator to produce sparks in Gaussian distribution. Under the effect of the two operators, if the produced spark is not in the feasible

region, the mapping rule will map the new generated sparks into the feasible region. To select the sparks for next generation, the selection strategy is used. Fireworks algorithm runs iteratively until it reaches the termination conditions [1].

## 2.2 FWA Principle

### 2.2.1 *Explosion Operator*

In initialization, FWA generates  $N$  fireworks randomly. Then the  $N$  fireworks generate sparks by explosion operations. The explosion operator is a key in FWA and plays an important role. The explosion operator include explosion strength, explosion amplitude and displacement operation [1].

#### 2.2.1.1 Explosion Strength

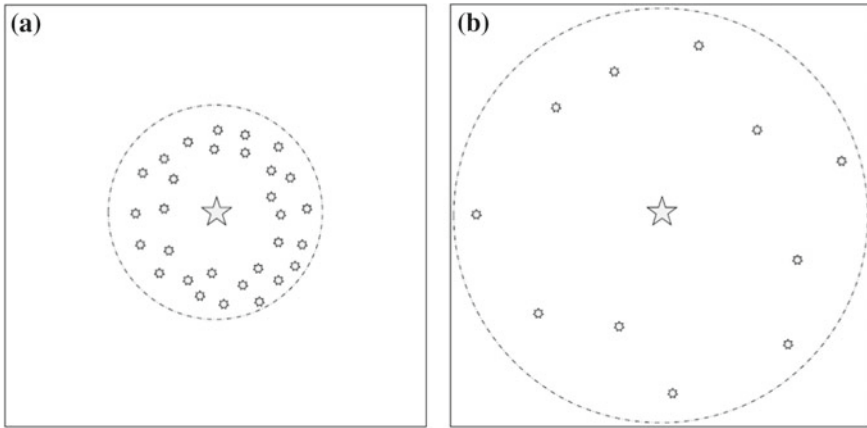
The explosion strength is a core operation in explosion operator. It simulates the way of explosion of fireworks in real life. When a firework blasts, the firework vanishes in one second and then many small bursts appear around it. Fireworks algorithm first determines the number of sparks, then calculates the amplitudes of each explosion.

Through the observations on the curves of some typical optimization functions, it can be seen that there are more points with good fitness values around the optima than those away from the optima. Therefore, fireworks with better fitness values produce more sparks, avoiding swing around the optima but fail to locate it. For fireworks with worse fitness values, their generated sparks are less in number and sparse in distribution, avoiding unnecessary computing. Fireworks with worse fitness values are used to explore the feasible space, preventing the algorithm from premature convergence. Fireworks algorithm determines the number and amplitude of the fireworks according to their fitness values, letting fireworks with better fitness values produce more sparks within a smaller amplitude and vice versa as shown in Fig. 2.1.

It can be seen from Fig. 2.1 that fireworks with better fitness values produce more sparks within a smaller amplitude (good explosion) than those with worse fitness values within a larger amplitude (bad explosion). After determining the number of sparks, it is needed to calculate the amplitude of the sparks in the explosion of a firework.

#### 2.2.1.2 Explosion Amplitude

Through observation on the curves of some typical optimization functions, the points around the local optima and global optima always have better fitness values. Therefore, by controlling the explosion amplitude, the amplitude of fireworks with better



**Fig. 2.1** The good and bad explosions of a firework. Reprinted from Ref. [1], with kind permission from Springer Science+Business Media. **a** Good explosion. **b** Bad explosion

fitness values gradually reduce, leading fireworks algorithm to find the local and global optima. By contrast, fireworks with worse fitness values explore the optima through a large amplitude. This is how the FWA controls the magnitude of the explosion amplitude.

### 2.2.1.3 Displacement Operation

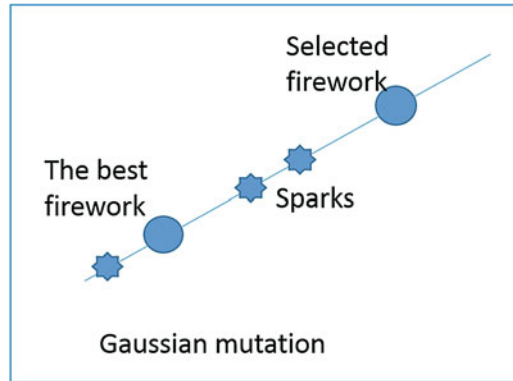
After the calculation of explosion amplitude, it is necessary to determine the displacement within the explosion amplitude. FWA uses the random displacement. In this way, each firework has its own specific explosion number and amplitude of sparks. FWA generates different random displacements within each amplitude to ensure the diversity of population. Through the explosion operator, each firework generates a shower of sparks, helping to find the global optimal of an optimization function.

## 2.2.2 Gaussian Mutation Operator

To further improve the diversity of a population, the Gaussian mutation is introduced into FWA. The way of producing sparks by Gaussian mutation is as follows: choose a firework from the current population, then apply Gaussian mutation to the firework in randomly selected dimensions.

For Gaussian mutation, the new sparks are generated between the best firework and the selected fireworks (Fig. 2.2). Yet, Gaussian mutation may produce sparks that exceed the feasible space. When a spark lies beyond the upper or lower boundary, the mapping rule will be carried out to map the spark to a new location within the feasible space.

**Fig. 2.2** Gaussian mutation. Reprinted from Ref. [1], with kind permission from Springer Science+Business Media



### 2.2.3 Mapping Rule

If a firework is near the boundary of the feasible space, while its explosion amplitude covers both the feasible and infeasible space, the generated sparks may lie out of the feasible space. As such, the spark beyond the feasible space is useless. Therefore, it needs to be getting back into the feasible space. The mapping rule is used to deal with this situation. The mapping rule ensures that all sparks are in the feasible space. If there is any spark that is generated by a firework beyond the feasible space, it will be mapped back to the feasible space.

### 2.2.4 Selection Strategy

After applying the explosion operator, the mutation operator, and the mapping rule, some of the generated sparks need to be selected and passed down to the next generation. The distance-based strategy is used in the fireworks algorithm. In order to select the sparks for next generation, first, the best spark is always kept for next generation. Then, the other  $(N - 1)$  individuals are selected based on distance maintaining the diversity of the population. The individual that is farther from the other individuals has greater chance to be selected than those individuals near the other individuals.

## 2.3 Implementation of FWA

FWA starts to run iteratively till the given termination conditions are met. It consists of the explosion operator, the mutation operator, the mapping rule, and the selection strategy. There are two termination conditions, such as meeting the accuracy requirements and reaching the maximum number of function evaluations.

The realization of FWA consists of four steps as follows.

- (1) Randomly generate fireworks in the feasible space.
- (2) Calculate the fitness value of each firework according to the fitness function. The number of sparks is calculated based on immune concentration theory in immunology and the fireworks with better fitness values produce more sparks.
- (3) Considering the fireworks phenomena in reality and the landscape of the functions, the fireworks generate sparks within a certain amplitude in FWA. The explosion amplitude is determined by the fitness value of that firework. The explosion amplitude for the firework with better fitness value is smaller and vice versa. Each spark represents a solution in the feasible space. To keep the diversity of the population, mutation operation is needed and Gaussian mutation is one of them.
- (4) Calculate the best fitness value. If the terminal condition is met, stop the algorithm. Otherwise, continue the iteration process. The best spark and the selected sparks formed a new population.

### 2.3.1 Explosion Operator

#### 2.3.1.1 Explosion Strength

In the explosion strength, i.e. the number of sparks is determined as follows:

$$S_i = m * \frac{Y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^N (Y_{\max} - f(x_i)) + \varepsilon}, \quad (2.1)$$

where  $S_i$  is the number of sparks for each individual or firework,  $m$  is a constant standing for the total number of sparks, and  $Y_{\max}$  means the fitness value of the worst individual among the  $N$  individuals in the population. Function  $f(x_i)$  represents the fitness for an individual  $x_i$ , while the last parameter  $\varepsilon$  is used to prevent the denominator from becoming zero.

The limitation of number of sparks are as follows:

$$\hat{s}_i = \begin{cases} \text{round}(a \cdot m), & \text{if } s_i < am \\ \text{round}(b \cdot m), & \text{if } s_i > bm, \ a < b < 1, \\ \text{round}(a \cdot m), & \text{otherwise} \end{cases} \quad (2.2)$$

where  $a$  and  $b$  are constant,  $\hat{s}_i$  is the limitation of the number of sparks, and  $\text{round}()$  is the rounding function.

### 2.3.1.2 Explosion Amplitude

The explosion amplitude is defined below.

$$A_i = \hat{A} * \frac{f(x_i) - Y_{\min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - Y_{\min}) + \varepsilon}, \quad (2.3)$$

where  $A_i$  denotes the amplitude of each individual,  $\hat{A}$  is a constant as the sum of all amplitudes, while  $Y_{\min}$  means the fitness value of the best individual among the  $N$  individuals. The meaning of function  $f(x_i)$  and parameter  $\varepsilon$  are the same as aforementioned in Eq. (2.2).

### 2.3.1.3 Displacement Operation

Displacement operation is to make displacement on each dimension of a firework and can be defined as

$$x_i^k = x_i^k + U(-A_i, A_i), \quad (2.4)$$

where  $U(-A_i, A_i)$  denotes the uniform random number within the intervals of the amplitude  $A_i$ .

Algorithm 2.1 is the pseudo code of the explosion operator described in Eqs. (2.1)–(2.4).

---

**Algorithm 2.1** Generate sparks

---

- 1: Initialization, calculate the fitness value  $f(x_i)$  for each firework.
  - 2: Calculate the number of sparks  $S_i$ .
  - 3: Calculate the amplitude of sparks  $A_i$ .
  - 4:  $z = \text{rand}(1, \text{dimension})$  //randomly choose  $z$  dimensions
  - 5: **for**  $k = 1 \rightarrow \text{dimension}$  **do**
  - 6:   **if**  $k \in z$  **then**
  - 7:      $x_i^k = x_i^k + U(-A_i, A_i)$
  - 8:   **end if**
  - 9: **end for**
- 

### 2.3.2 Mutation Operator

Suppose the position of current individual is stated as  $x_i^k$ , where  $i$  varies from 1 to  $N$  and  $k$  denotes the current dimension. The sparks of Gaussian explosion are calculated by

$$x_i^k = x_i^k * g, \quad (2.5)$$

where  $g$  is a random number in Gaussian distribution with mean 1 and variance 1 such as

$$g = \mathcal{N}(1, 1). \quad (2.6)$$

Algorithm 2.2 shows the pseudo code for Gaussian mutation.

---

**Algorithm 2.2** Gaussian Mutation

---

```

1: Calculate the fitness value  $f(x_i)$  for each firework.
2: Calculate the coefficient  $g = \mathcal{N}(1, 1)$ .
3:  $z = \text{rand}(1, \text{dimension})$  //randomly select  $z$  dimensions
4: for  $k = 1 \rightarrow \text{dimension}$  do
5:   if  $k \in z$  then
6:      $x_i^k = x_i^k * g$ 
7:   end if
8: end for

```

---

### 2.3.3 Mapping Rule

The mapping rule ensures all the individuals stay in the feasible space. If there are some outlying sparks from the boundary, they will be mapped back to their allowable scopes.

The mapping rule utilizes a modular operation and is stated as follows:

$$x_i^k = X_{LB,k} + x_i^k \% (X_{UB,k} - X_{LB,k}), \quad (2.7)$$

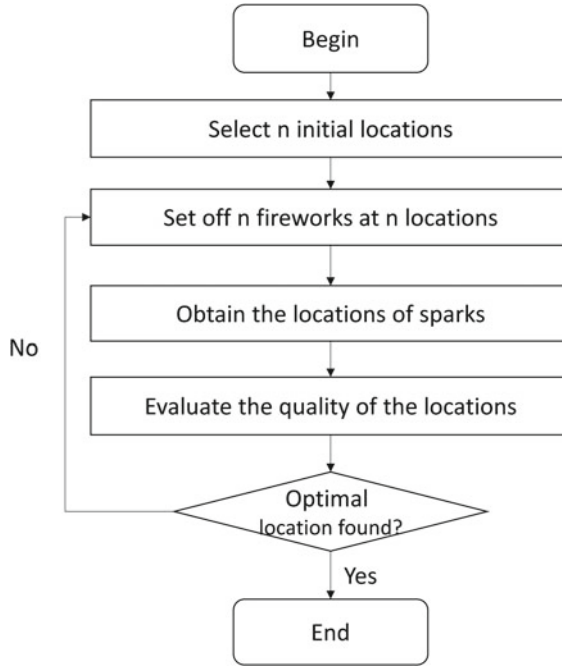
where  $x_i^k$  represents the positions of any sparks that lie out of bounds, while  $X_{UB,k}$  and  $X_{LB,k}$  stand for the maximum and minimum boundaries of a spark position. The sign  $\%$  represents modular arithmetic.

### 2.3.4 Selection Strategy

In selection strategy, the measurement of Euclidean distance is used, where  $d(x_i, x_j)$  denotes the Euclidean distance between any two individuals  $x_i$  and  $x_j$ .

$$R(x_i) = \sum_{j=1}^K d(x_i, x_j) = \sum_{j=1}^K \|x_i - x_j\|, \quad (2.8)$$

where  $R(x_i)$  represents the sum of distances between individual  $x_i$  and all the other individuals.  $j \in K$  means the position  $j$  belongs to set  $K$ , where  $K$  is the set of



**Fig. 2.3** The flowchart of FWA

combining both the sparks generated by explosion operator and mutation operator. The roulette way is used to choose individuals for next generation, as the possibility for choosing the individual  $x_i$  should be  $p(x_i)$ , which is given by

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)}. \quad (2.9)$$

From Eq.(2.9), it can be seen that individuals with larger distances will have more chances to be selected for next generation. In such a way, the diversity of the population can be guaranteed.

The flowchart of FWA is depicted in Fig. 2.3.

The pseudo code of FWA is shown in Algorithm 2.3.



**Algorithm 2.3** Pseudo code of FWA

---

```

1: Randomly select  $N$  locations for fireworks
2: while terminal condition is not met do
3:   Set off  $N$  fireworks, respectively, at the  $N$  locations:
4:   for all fireworks  $x_i$  do
5:     Calculate the number of sparks as  $S_i$ 
6:     Calculate the amplitude of sparks as  $A_i$ 
7:   end for
   //  $\hat{m}$  is the number of sparks generated by Gaussian mutation
8:   for  $k = 1 \rightarrow \hat{m}$  do
9:     Randomly select a firework  $x_i$  and generate a spark
10:   end for
11:   select the best spark and the other sparks according to selection strategy
12: end while

```

---

**2.4 The Characteristics of FWA**

FWA contains the following characteristics: explosion, instantaneity, simplicity, locality, emergent property, distribute parallelism, diversity and extendibility. The details are given below.

**2.4.1 Explosion**

After the first iteration of FWA, the fireworks explode within the amplitude and produce a shower of sparks. At the end of the iteration, the algorithm selects  $N$  sparks for the next generation. The  $N$  selected sparks are treated like new fireworks, preparing for explosion in the next iteration. In each iteration, the fireworks will explode, indicating the explosive characteristic of FWA.

**2.4.2 Instantaneity**

In each iteration, FWA calculates the number of sparks and the explosion amplitude, depending on the fitness values of the fireworks. Then, the sparks are produced by the explosion and mutation operators. Finally, the best spark is preserved at first and then the other  $(N - 1)$  sparks are selected based on a selection strategy. The selected  $N$  sparks are treated as the fireworks for the next generation, while the rest of the sparks are no longer reserved. Sparks or fireworks are not kept, indicating the instantaneous characteristic of FWA.

### ***2.4.3 Simplicity***

Like other swarm intelligence algorithms, each firework in FWA only percepts its own information itself and its surrounding information, following simple rules to complete their missions. Overall, FWA is not complex, composed of simple individuals. Therefore, FWA is characteristic of simplicity.

### ***2.4.4 Locality***

In FWA, all the fireworks generate sparks within their amplitudes. Unless beyond the feasible region, sparks are confined within a certain range. Localized features of FWA reflect the powerful local search capabilities, as the algorithm can be used for local search in the latter of the search process. Therefore, FWA contains locality.

### ***2.4.5 Emergent Property***

Fireworks are in competition and collaboration with each other and the group showed high degree of intelligence which a simple individual cannot achieve. Interaction between fireworks are more complicated than a single individual's behavior. Therefore, firework algorithm has the characteristic of emergent.

### ***2.4.6 Distributed Parallelism***

In each iteration of FWA, each firework explodes and searches within different space, i.e., each firework conducts a search in different dimensions. Finally, the sparks and fireworks are combined together to choose  $N$  fireworks for the next generation. In each iteration, FWA searches the space in parallel, showing the characteristic of distribution parallelism.

### ***2.4.7 Diversity***

Population diversity is vital to the performance of any swarm intelligence algorithm. By maintaining the population diversity, the algorithm can jump out of local optima, which makes the algorithm converge to the global optimal point, which a generic optimization can hardly achieve. Therefore, swarm optimization algorithms are different from any generic optimization algorithm. The better the population diversity

is, the wider the individuals are distributed. The optimal value might be easier to be found if a population is strongly diverse, as the convergence of the algorithm will not be affected significantly. Thus, population diversity is an important part of the FWA. The diversities of FWA can be concluded as follows.

#### **2.4.7.1 The Diversity of the Number of Sparks and Explosion Amplitude**

According to the explosion operator and the fitness values, each firework generates a different number of sparks within a different magnitude. The fireworks with higher fitness values produce more sparks within smaller ranges, while the fireworks with lower fitness values produce fewer sparks within larger ranges. In such a way, the diversity of the population would be guaranteed.

#### **2.4.7.2 The Diversity of Displacement and Gaussian Mutation**

FWA has two operators, explosion operator and mutation operator. In the explosion operator, a displacement is calculated according to an amplitude. The displacement is added to a position in a dimension of a selected firework. In the Gaussian mutation operator, the selected fireworks need to multiply a Gaussian random number in the position of a dimension. The explosion operator is relative to the fitness values of fireworks while the mutation operator is relative to the position of the fireworks. The two operators are different from each other, but both of them guarantee the explosion to be diverse.

#### **2.4.7.3 The Diversity of Fireworks**

Through a certain selection mechanism, the coordinates of the retained fireworks are different. As a result, these phenomena ensure the diversity of the population. In addition, in the selection strategy, sparks with a greater distance from the other sparks are more likely to be selected, which in turn, ensures the diversity of population.

### **2.4.8 Extendibility**

In FWA, the number of sparks are uncertain and able to be determined based on the complexity of the problem in hand. The number of fireworks and sparks can be more or less, as both increase and decrease of the individuals can effectively solve the problem. Therefore, FWA has extendibility.

### 2.4.9 Adaptability

When solving problems using FWA, it is unnecessary for the problem to be of an explicit expression. The problem can be solved by calculating the fitness values only. Meanwhile, FWA can also solve the problems with explicit expressions, indicating its capability. Therefore, FWA is of adaptability and can be regarded as an adaptive algorithm.

## 2.5 Impact of Operators in FWA on Performance

### 2.5.1 Explosion Operator

When a spark explodes, the area around the spark is searched. If the fitness value of a spark is higher, the amplitude of the spark is larger and the number of small bursts of the spark is fewer. In this case, the sparks with better fitness values will search more carefully in smaller areas, while the sparks with worse fitness values will search in wider areas. Therefore, FWA has greater chances to find the global optimal in limited function evaluations.

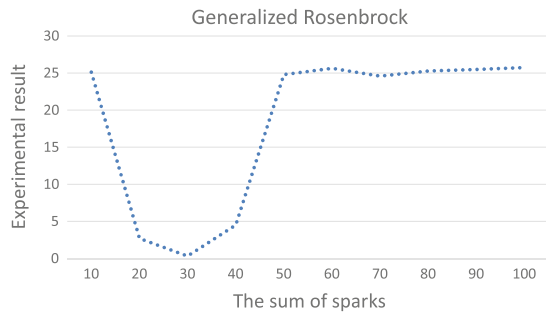
The explosion operator has two parameters. The first parameter  $m$  is used to limit the total number of sparks and the second parameter  $N$  is the number of fireworks.

Function Generalized Rosenbrock is used to illustrate the impact of the explosion operator on the performance of FWA.

Figure 2.4 gives the impact of the total number of sparks  $m$  on the performance of FWA on function Generalized Rosenbrock, under the circumstance that the other parameters remain unchanged.

Experimental results show that better performances are obtained when the total number of sparks  $m$  is set between 10 and 50, while the other parameters remained unchanged.

**Fig. 2.4** The impact of different number of sparks on the performance of FWA for Generalized Rosenbrock function. The vertical axis represents the accuracy of the experimental results



**Fig. 2.5** The impact of the number of fireworks on the performance of FWA for Generalized Rosenbrock function. The *vertical axis* represents the accuracy of the experimental results

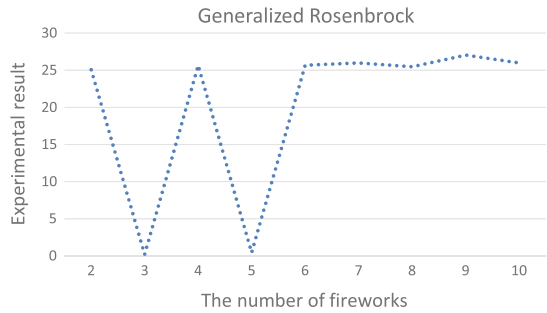


Figure 2.5 gives the impact of the total number of fireworks  $N$  on the performance of FWA on Generalized Rosenbrock function.

It can be seen from Fig. 2.5 that better performances are obtained when the total number of fireworks  $N$  is set as 3 or 5, while the other parameters remain unchanged.

Obviously, for different optimization problems, setting different values of parameters can have a certain impact on the performance of the algorithm. Better performances on function Generalized Rosenbrock are obtained when the total number of sparks is set between 20 and 40, and 3 or 5 fireworks if other parameters remain unchanged.

2.5.2 Gaussian Mutation

Gaussian mutation operator can increase the diversity of the algorithm because the sparks generated by Gaussian mutation are not limited to the area around the fireworks. In addition, since the sparks are generated between the current location and the origin by Gaussian mutation ( $x_i^k \leq x_i^k * g$ ), the performance is pretty good for functions having the optimal at the origin. For example, FWA with Gaussian mutation can easily find the location of the optimal value on function Sphere because the optimal value of function Sphere is at the origin.

Table 2.1 gives the experimental results of FWA on two functions. The functions are 30-dimensional and the algorithm iterates 300,000 times.

**Table 2.1** FWA with and without Gaussian mutation on function Sphere and Generalized Rosenbrock

Status	Sphere	Generalized Rosenbrock
FWA with Gaussian mutation	0	25.209447
FWA without Gaussian mutation	1.095037	706.936069

It can be seen from Table 2.1 that the performances of FWA on both functions are greatly improved by Gaussian mutation operator. This is due to the diversity of the population increased by Gaussian mutation. The global search ability of FWA is enhanced.

### 2.5.3 Mapping Rule

The mapping rule is used to ensure that all the sparks are generated within the scope of the feasible space. When a firework is near the border and the amplitude of the explosion is large, the generated sparks might be out of the boundaries of the feasible space. Therefore, the sparks that are out of the boundaries will be mapped into feasible space for avoiding unnecessary computation. However, the mapping rule has its own disadvantages. For instance, it can easily pull a spark to the locations near the origin, benefiting the functions with optima near the origin.

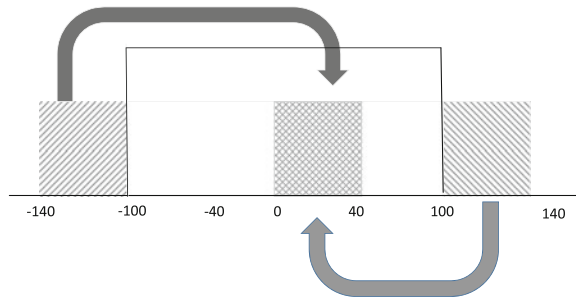
The mapping rule adopts a modular arithmetic to ensure that the out-of-range sparks are pulled back into the feasible space.

As shown in Fig. 2.6, the feasible space is set from  $-100$  to  $100$  and the limitation of explosion amplitude is set as  $40$ . Therefore, the points beyond the boundaries will fall into the area of  $-140 \sim -100$  and  $100 \sim 140$  shown as shadow areas in Fig. 2.6. According to the mapping rule, points located in these two areas are mapped into the range from  $0$  to  $40$ , which is near the origin.

### 2.5.4 Selection Strategy

The selection strategy is to choose individuals for the next generation. The best individual is always kept for the next generation, while the remaining  $(N - 1)$  individuals are selected based on Euclidean distance and the sparks further away from other sparks can be selected with larger possibilities. Thus, the diversity of FWA is guaranteed in such a way.

**Fig. 2.6** The operation of mapping rule



## 2.6 Comparison of FWA with Three Other SI Algorithms

### 2.6.1 Ideas Comparison Between FWA and GA

The idea of immune density is introduced into the selection strategy of FWA, so it is unnecessary to design a new selection operator. Selection strategy in FWA sounds like the selection operator in genetic algorithm, but they are different. Based on the idea of immune density, each spark in FWA is treated as an antibody in immune system. A spark (antibody) which has more similar sparks (antibodies) is chosen with a lower probability. On the contrary, a spark (antibody) which has less similar sparks (antibodies) is selected with a higher chance. Hence, the sparks (antibodies) with lower fitness values have the chance to be selected such that the diversity of the sparks (antibodies) is ensured. Compared with the genetic algorithm, the selection operator determines which individual is to be selected by the fitness values of the individuals. The selection is based on the roulette and the diversity of the population is not guaranteed.

Genetic algorithm was originally proposed by Prof. Holland of the University of Michigan [2]. At that time, Holland recognized that biological hereditary and natural evolution phenomena are similar to artificial adaptive systems. The idea of genetic and evolutionary in nature could be used to study the generation of natural, artificial adaptive systems and their relationship with the environment. He suggested to use the mechanism of genetic in study and design of artificial adaptive systems, as the swarms could be used for adaptive search and the crossover and mutation operations are vital.

The common aspects of both algorithms are as follows:

- (1) Randomly initialize the initial population;
- (2) Calculate the fitness value of each individual;
- (3) A series of operations to be performed according to fitness values, such as selection operator, crossover operator, and mutation operator in genetic algorithm, and explosion operator and mutation operator in FWA;
- (4) Select the individuals for the next generation according to the fitness values;
- (5) Stop when the termination conditions are met. Otherwise, go to step 2.

From the above steps, we can see that the FWA and the genetic algorithms have a lot in common. Both randomly initialize a population, evaluate the individuals according to their fitness values and perform a certain random search. In addition, both algorithms are not guaranteed to find the optimal values.

There is no crossover operator in the FWA and the mutation operator in the FWA is totally different from that in the genetic algorithm.

Compared with the genetic algorithm, information sharing mechanisms in FWA is quite different. In the genetic algorithm, chromosomes share information with each other so the entire population moves relatively homogeneously in the feasible space. However, in the FWA, a distributed information sharing mechanism is used, while the number of sparks and the explosion amplitudes are determined by the fitness

values of fireworks which are located in different areas. In addition, the fireworks are always selected from different areas and can hardly stay together due to the immune-based selection. Yet, the FWA has more mechanisms to avoid premature compared to genetic algorithm.

### ***2.6.2 Ideas Comparison Between FWA and Two Versions of PSO***

Two particle swarm optimization algorithms are introduced at first, i.e., clonal particle swarm optimization (CPSO) [3] and standard particle swarm optimization (SPSO) [4].

In the biological immune system, when the antigen gets into a living body, the immune system in the body can identify and eliminate the antigen. This process is mainly by cloning that activates antibodies, increasing their number and clearing the antigens [5]. Based on realizing the importance of the immune response, Tan and Xiao [3] made improvements to the standard PSO algorithm by proposing a cloning operator.

The process of the three algorithms is similar as explained below.

- (1) Initialization. FWA initializes the fireworks, while the two kinds of PSO initialize particles.
- (2) Calculate the fitness values for the individuals in Step 1.
- (3) Process necessary operations. FWA processes the explosion and mutation operations, while the two kinds of PSO processes update pbest and gbest. Also, the position and speed for each particle need to be updated in each generation.
- (4) Select individuals for the next generation.
- (5) If the termination condition is met, the algorithm is terminated. Otherwise, go to Step 2.

From the above steps, we can see the FWA and the two kinds of PSO have much in common. They adopt random initial populations, evaluate the functions and perform the search based on the fitness values. Also, all the algorithms are not guaranteed to find the optimal solution.

However, there is no mutation operation in SPSO, while there is Gaussian mutation in CPSO and both displacement operation and Gaussian mutation in FWA. Furthermore, the Gaussian mutation in FWA plays a role on differential dimensions with the same displacement, making connections between the different dimensions. The displacement in CPSO differs in each dimension. Besides, Gaussian mutation takes place in each iteration in FWA, but runs once in several iterations in CPSO.

Compared with the two kinds of PSO, information sharing mechanism in FWA is different. In the two kinds of PSO, only gbest gives information to other particles, which is one way of information delivery, as the search process follows the information about the best particle. FWA, on the other hand, uses a distributed information sharing mechanism, so as to determine the number of sparks and explosion amplitude



by the fitness values of each spark in different regions. It also needs to maintain the best firework throughout the iterative process.

Besides, FWA utilizes the idea of immune concentration to keep the diversity of the population, whereas the idea is not contained in SPSO.

## 2.7 Experimental Results and Analysis

### 2.7.1 Benchmark Functions

FWA [1] and SPSO [4] and CPSO [3] are compared on six test functions. The details of the six test functions can be seen in Appendix A.

### 2.7.2 Parameters Setting

- The population size is set as 5 and the spark of Gaussian mutation is set as 5.
- The total number of sparks is set as 50. Parameter  $a$  and  $b$  are set as 0.8 and 0.04.
- The constant  $\hat{A}$  is set as 40. There is no lower boundary for the amplitude of explosion.
- Function is 30 dimensions, running 20 times and the number of function evaluations is limited to 400,000.

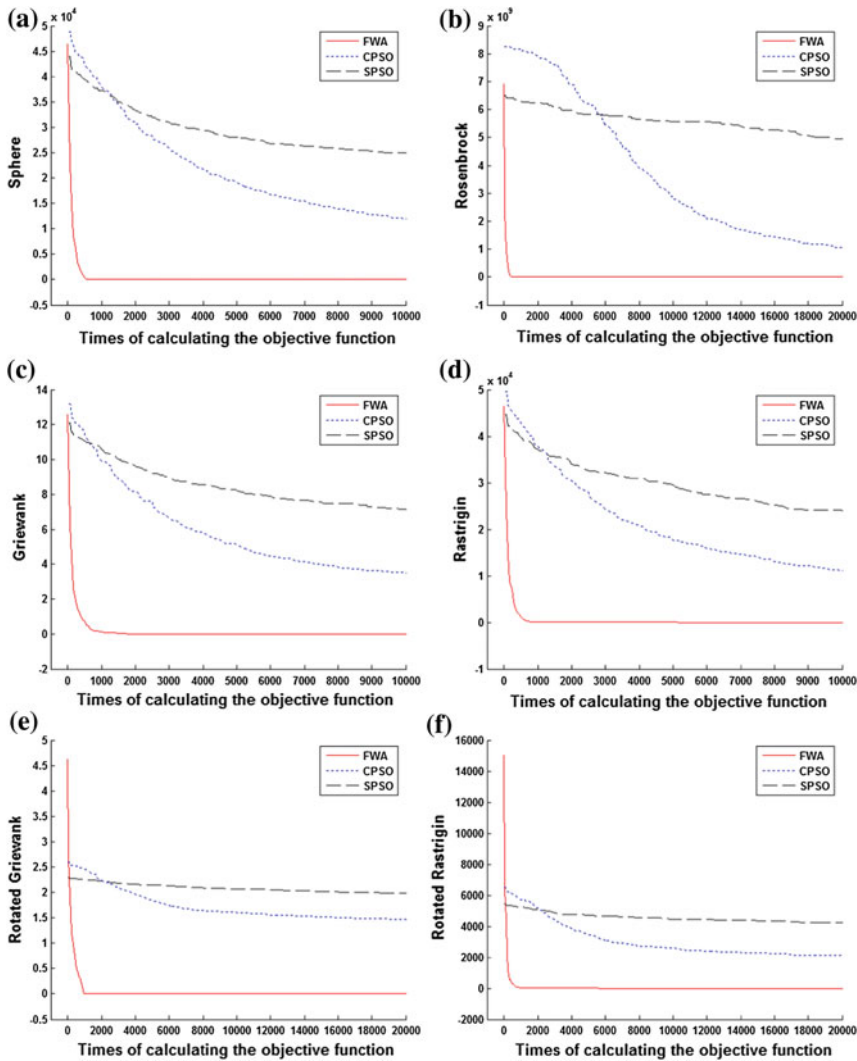
### 2.7.3 Experimental Results

The experimental results are shown in Table 2.2 when the functions are evaluated for 400,000 times (the accuracy is  $10^{-6}$ ).

The convergence curves are also shown in Fig. 2.7.

**Table 2.2** Comparison of FWA with CPSO and SPSO (lower than  $10^{-6}$  is treated as zero)

	FWA		CPSO		SPSO	
	Mean	Std	Mean	Std	Mean	Std
Sphere	0	0	0	0	367.1166	186.7949
Rosenbrock	12.16293	12.82113	66.58722	204.2907	5692076	4087432
Griewank	0	0	0.003693	0.011792	1.088648	0.042218
Rastrigin	0	0	6.769299	7.701368	676.1549	197.9695
Rotated Griewank	0	0	0.043401	0.042286	0.920613	0.088088
Rotated Rastrigin	0	0	23.92579	13.6093	339.2073	62.38145



**Fig. 2.7** Convergence curves of FWA, CPSO and SPSO. Reprinted from Ref. [1], with kind permission from Springer Science+Business Media

### 2.7.4 Analysis

It can be seen from the experimental results that FWA works significantly better than SPSO and CPSO [3] in both convergence speed and accuracy on the six test functions. Hence, FWA has good convergence and result accuracy and can be successfully applied to function optimization problems. In the comparison experiments with SPSO and CPSO, FWA reveals good advantage, which means FWA is very successful and has good prospects. In addition, since a lot of practical engineering problems can

be transformed into the function optimization problem and FWA can solve function optimization problems effectively, FWA has good prospects.

However, studies on FWA is still in its infancy, while current FWA still have some problems. Further research can be concluded as follows.

- (1) The main operators of FWA are realized, but they are still not perfect. For instance, Gaussian mutation achieves good performance, but it cannot solve all the problems effectively. New mechanism needs to be added to improve FWA.
- (2) Parameters of the algorithm are set according to empirical experiments on simple functions. The experiments are not enough and there is no theoretical analysis for the settings. The future research is to find reasonable parameters for each problem.
- (3) The function optimization experiments are conducted on a few benchmark functions. Hence, the experimental results are not convincing in some aspects. More optimization functions are needed to make the test systematic and comprehensive.
- (4) FWA can be applied to many areas. For example, neural networks, fuzzy systems control and fuzzy rule learning. Yet, FWA can be applied to solving discrete issues.

## 2.8 Summary

This chapter described FWA in detail, including explosion operator, mutation operator, mapping rule and selection strategy. The flowchart and pseudo code of FWA were also given while the characteristics of FWA and the impact of various factors on the performance of FWA were analyzed deeply. Besides, GA, PSO, and FWA were compared. Experimental results show that, FWA had better performance compared to GA and PSO.

FWA and its applications prove that it can solve many complex optimization problems effectively. Furthermore, FWA can be parallelized and thus is suitable for dealing with big data problems. Whether for theoretical or applied researches, FWA is worth researching and might bring great scientific and economic benefits for us.

## References

1. Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in *Advances in Swarm Intelligence* (Springer, Berlin, 2010), pp. 355–364
2. J.H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. (U Michigan Press, 1975)
3. Y. Tan, Z.M. Xiao. Clonal particle swarm optimization and its applications, in *IEEE Congress on Evolutionary Computation, CEC 2007*, (2007), pp. 2303–2309
4. J. Kennedy, R. Eberhart et al., Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4(2) (Perth, Australia, 1995), pp. 1942–1948
5. R. Liu, H. Du, L. Jiao, Immunity poly-clonal strategy. *J. Comput. Res. Dev.* **41**(4), 571–576 (2004)

Fireworks Algorithm

A Novel Swarm Intelligence Optimization Method

Tan, Y.

2015, XXXIX, 323 p., Hardcover

ISBN: 978-3-662-46352-9