Applications of Improved Swarm Intelligence Algorithms in Complex Engineering Problems

Tao Zhang
Tianjin University

PART ONE

Swarm Intelligence and Complex Engineering Problems

1.1 Basic Concepts of Evolutionary Computation

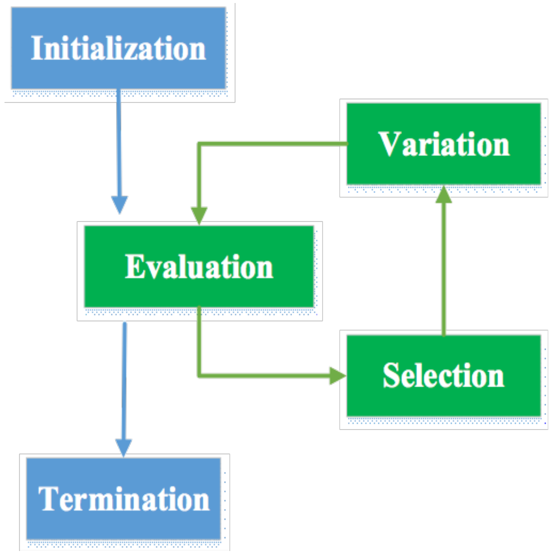


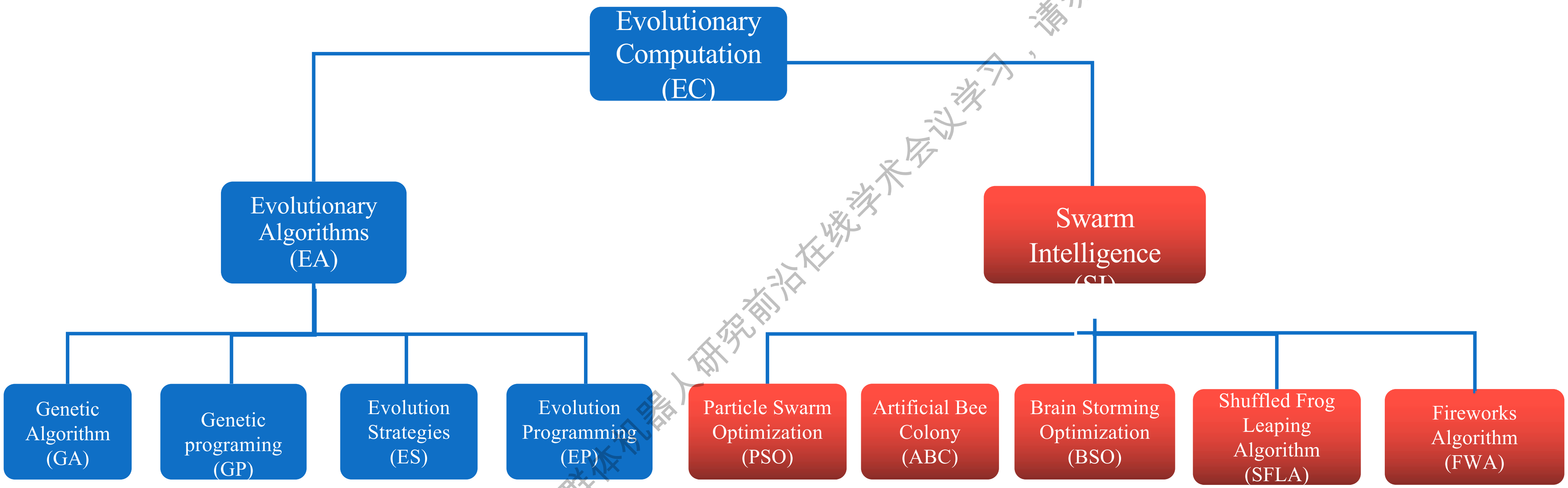
Fig. 2 Main Steps of Evolutionary Computation

Evolutionary Computation was inspired and derived from natural evolution or group behaviors.

It has three main steps as follows:

- **Initialization: initialize the population**
- **Updating：**
  - ➢ **Evaluation：fitness evaluation for each individual**
  - ➢ **Selection：select individuals to form a new population**
  - ➢ **Variation：to generate new individuals**
- **Termination：output the final solution**

1.2 Classification of Evolutionary Computation



Natural Evolution **Group Behavior**

Fig. 3 Main Algorithms of Evolutionary Computation

The searching process of a swarm intelligence algorithm imitates group behavior. For example, artificial bee colony algorithm imitates the foraging behavior of bees. By imitating the behavior, the optimization solution can be quickly searched.

Shuffled frog leaping algorithm

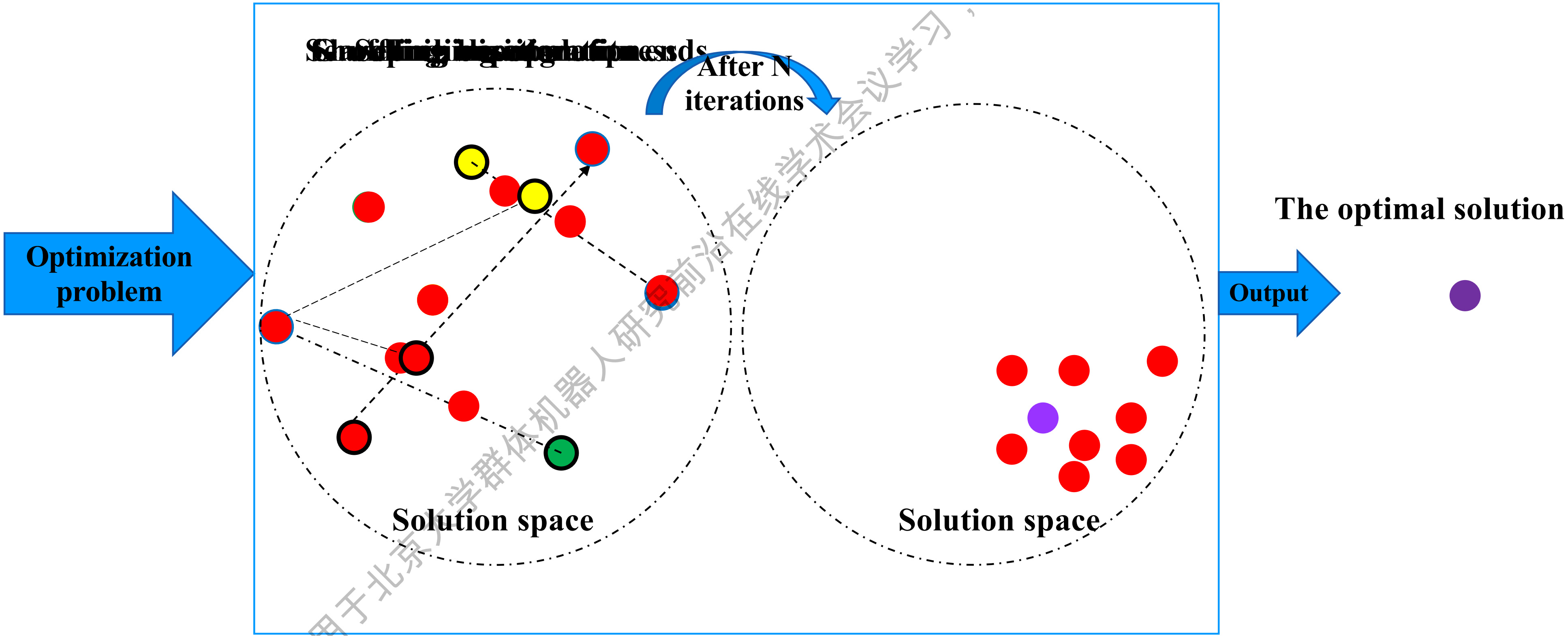


Fig. 4 Shuffled Frog Leaping Algorithm

FWA(Fireworks Algorithm)

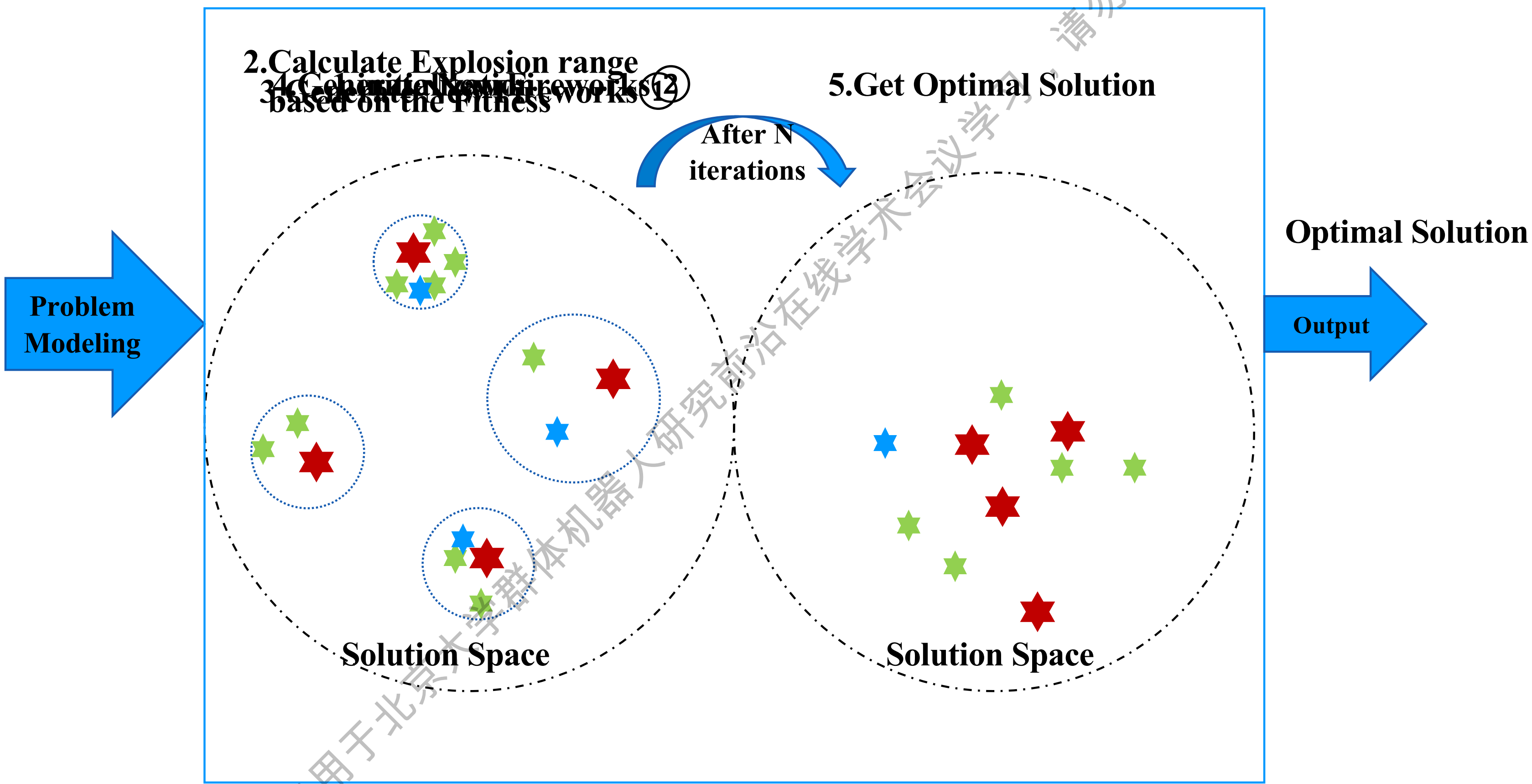


Fig. 5 Fireworks Algorithm

1.3 Blind Optimization

- We divide optimization algorithms into two categories: non-blind optimization algorithms and blind optimization algorithms.
- Non-blind optimization algorithms: The optimizing process is guided by the rules. Typically they are related to the objective functions, constraints or other conditions.
- Blind optimization algorithms:The optimizing process can be regarded as a black box. Wide range of applications and strong global search ability.
- SIs are blind optimization algorithms.

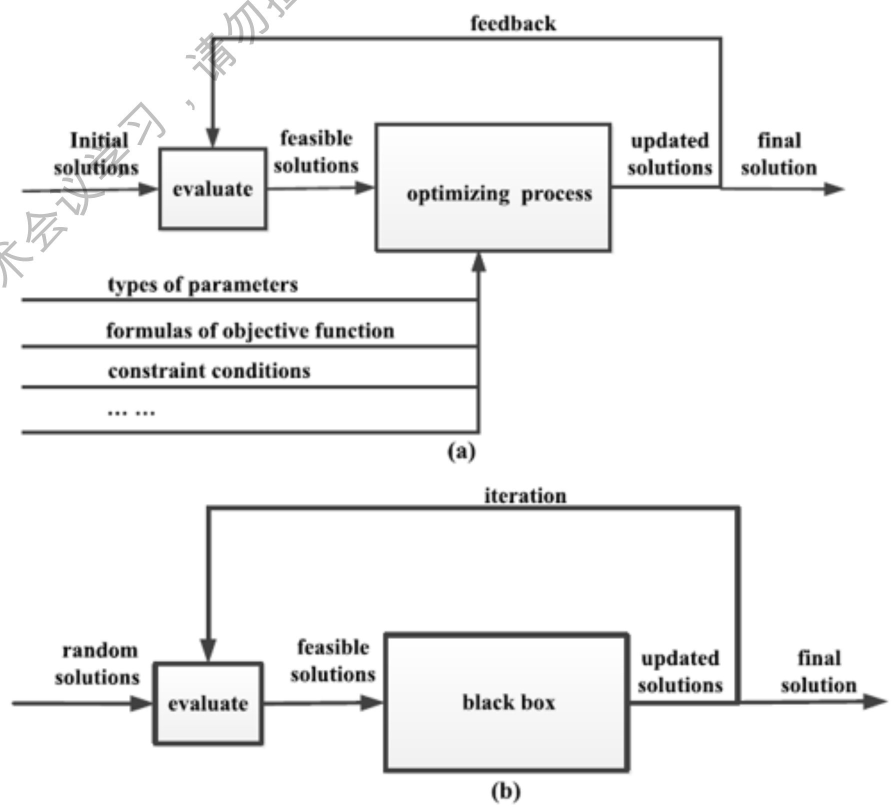


Fig. 6 Blind/Non-blind Optimization Algorithm
(a)Non-blind optimization algorithm
(b)blind optimization algorithm

Complex Engineering Problems

- In practice, a lot of engineering applications need to be optimized. The optimization problems for engineering applications are usually **nonconvex or multimodal, large-scale, highly constrained, multi-objective, and subject to a large number of uncertainties.**
- Because of the high complexity, most of complex engineering problems can not be solved effectively by non-blind optimization algorithms. The blind optimization algorithms can be applied.
- Three cases of complex engineering problems: **hardware/software partitioning, feature selection and local backlight dimming.**

PART
TWO

Application of Swarm Intelligence

Applying SIs to Solve Complex Engineering Problems

Steps:

- ➢**Build the mathematical model of the engineering problem**
- ➢**Determine the solution space and fitness evaluation function.**
- ➢**Apply the SI to search the optimal solution.**
- ➢**Output the obtained best solution (the algorithm ends).**

CASE ONE

Hardware/Software Partitioning

Multi-Core Embedded System

The multi-core embedded systems have become the main platform for processing complex tasks. A multi-core embedded system is usually composed of software units and hardware units.

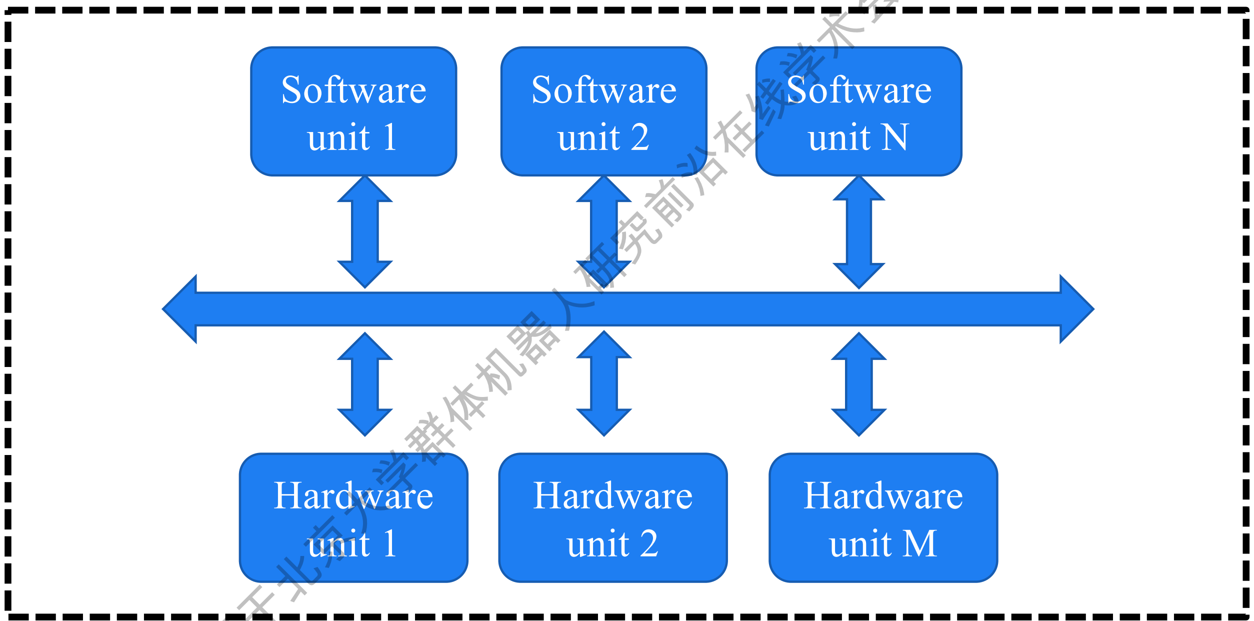


Fig. 7 Multi-core Embedded System Architecture

Description of Hardware/Software Partitioning

Hardware/software (HW/SW) partitioning:

A complex task can be divided into many subtasks.

The subtasks are assigned to different processing units.

The objective of hardware/software partitioning is to find the best task assignment scheme.

The subtasks to be assigned can be described by Fig.1(a). Where each subtask includes three attributes:

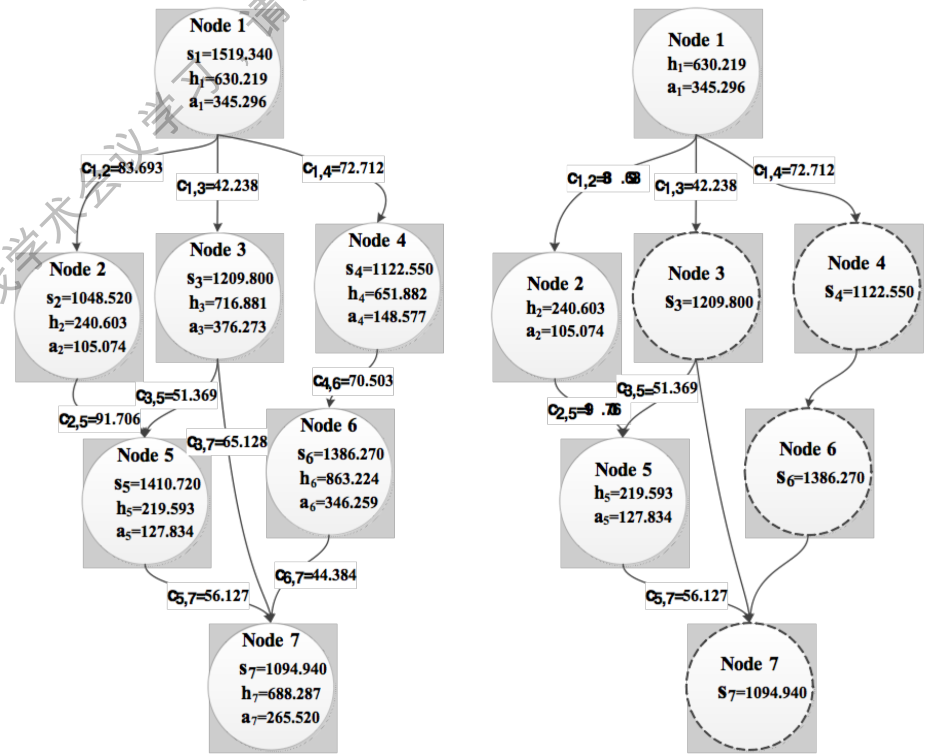
① software execution time s_i

② hardware execution time h_i

③ hardware area a_i

$c_{i,j}$ is the communication time between node *i* and *j*

Fig.1(b) shows a result after partitioning, where node 3, 4, 6 and 7 are assigned to software units and the other nodes are assigned to hardware units.



(a)Sub-tasks to be assigned (b) After HW/SW partitioning

Fig.8 Before and after HW/SW partitioning

Mathematical Model of Hardware/Software Partitioning

Assuming the optimization objective is minimizing the critical path which demonstrates the longest path, where the critical path determines the time required to execute the tasks on the embedded platform. The hardware area is set as the constraint. Then the optimization problem can be expressed by:

$$\min: T = \max\{TE(k) \mid 0,1 \dots M\}$$

$$subject\ to: \sum_{i=1}^{N} x_i \times a_i \leq A\_limit$$

Where

***TE*(*k*) is the completion time of the *k*-th path**

***M* is the number of paths**

***A_limit* is the constraint value of hardware area**

A partitioning scheme can be encoded to an *N*-dimensional vector, where *N* is the number of task nodes, $X = (x_1, x_2, \dots, x_N)^T$ $x_i \in \{0,1\}$

$x_i = 0$ **represents subtask *i* is assigned to the software,**

$x_i = 1$ **represents task *i* is assigned to hardware.**

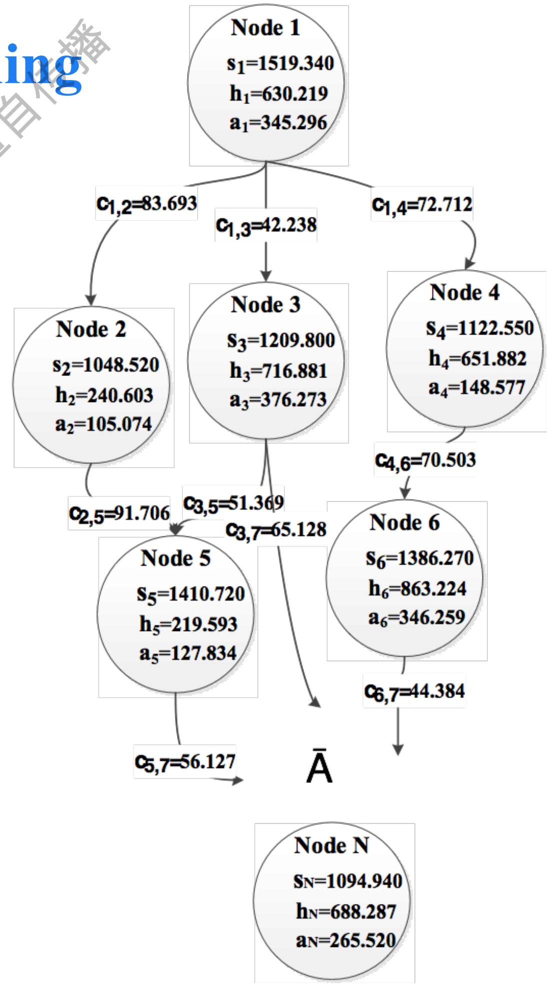


Fig. 9 Sub-tasks to be assigned

Hardware/Software Partitioning Based on SFLA-GRA

➢ **Shuffled Frog Leaping Algorithm——SFLA**

In this algorithm, the position of a frog represents a solution and the searching for the optimal solution is fulfilled on multiple iterations.
SFLA has strong global search ability to solve hardware/software partitioning problems.

➢ **GReedy Algorithm——GRA**
GRA was inspired from the greedy idea. It has a fast descending speed while it is easy to fall into local optimum.

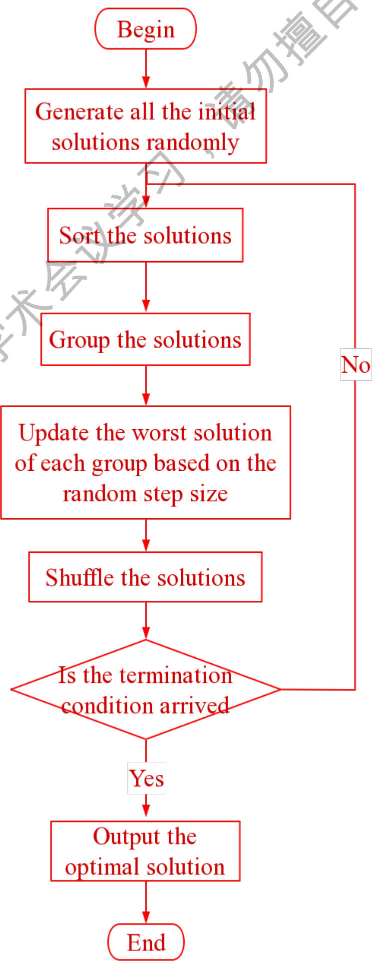


Fig. 10 The flow chart of SFLA

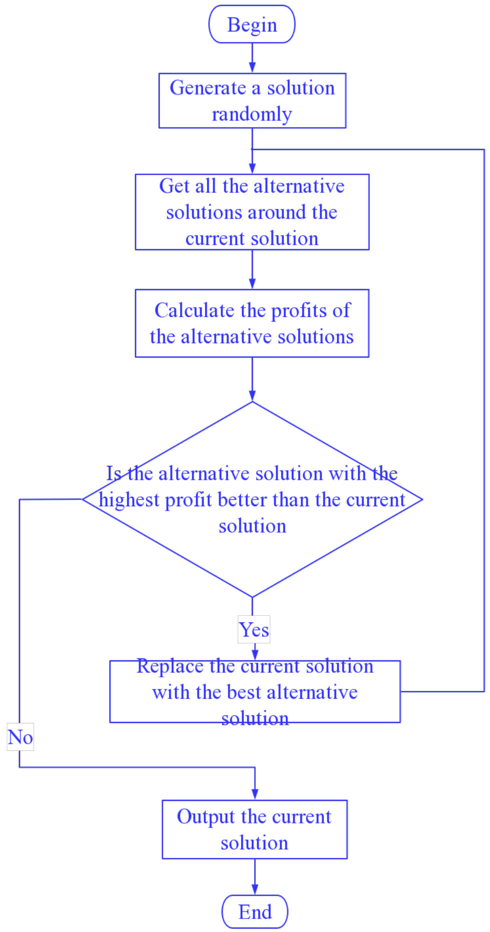


Fig. 11 The flow chart of GRA

Hardware/Software Partitioning Based on SFLA-GRA

➢ **Fusion of multiple algorithms——SFLA-GRA**

In order to utilize the advantages of the above two algorithms, we investigate SFLA which has strong global optimization ability and GRA has a high efficiency to design a new fusion algorithm, namely SFLA-GRA.

1) Terminate invalid iterations of SFLA with GRA.

2) Terminate the algorithm with a new termination condition.

3) Accelerate the search with greedy step size.

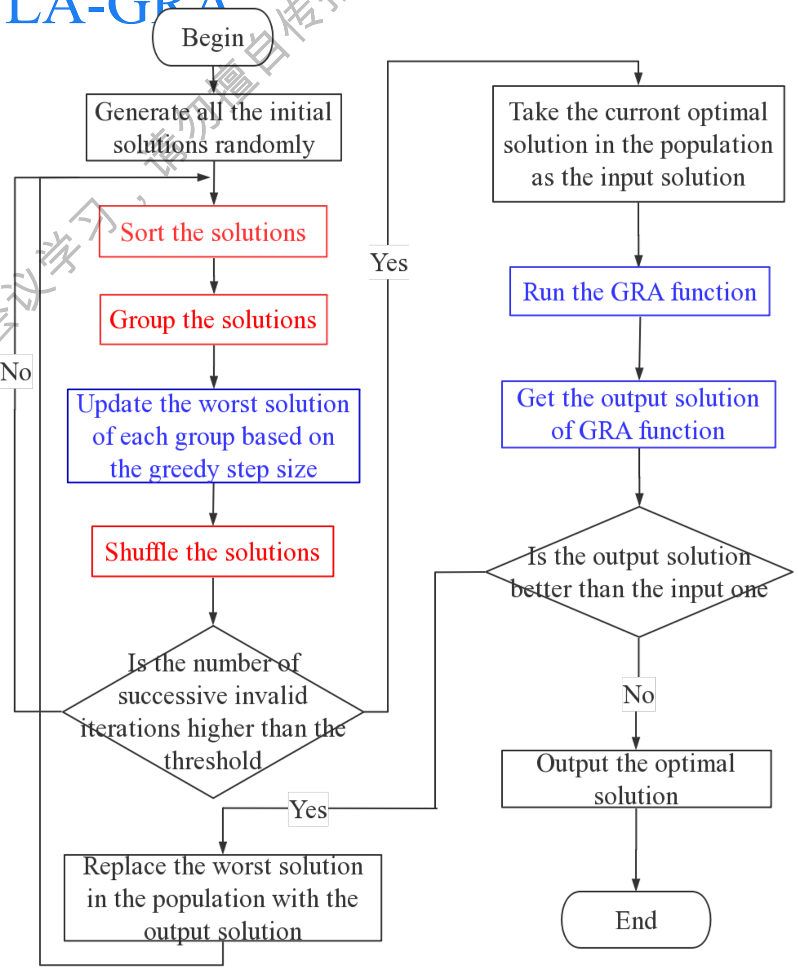


Fig. 12 The flow chart of SFLA-GRA

Experiment Results
of Hardware/Software Partitioning Based on SFLA-GRA

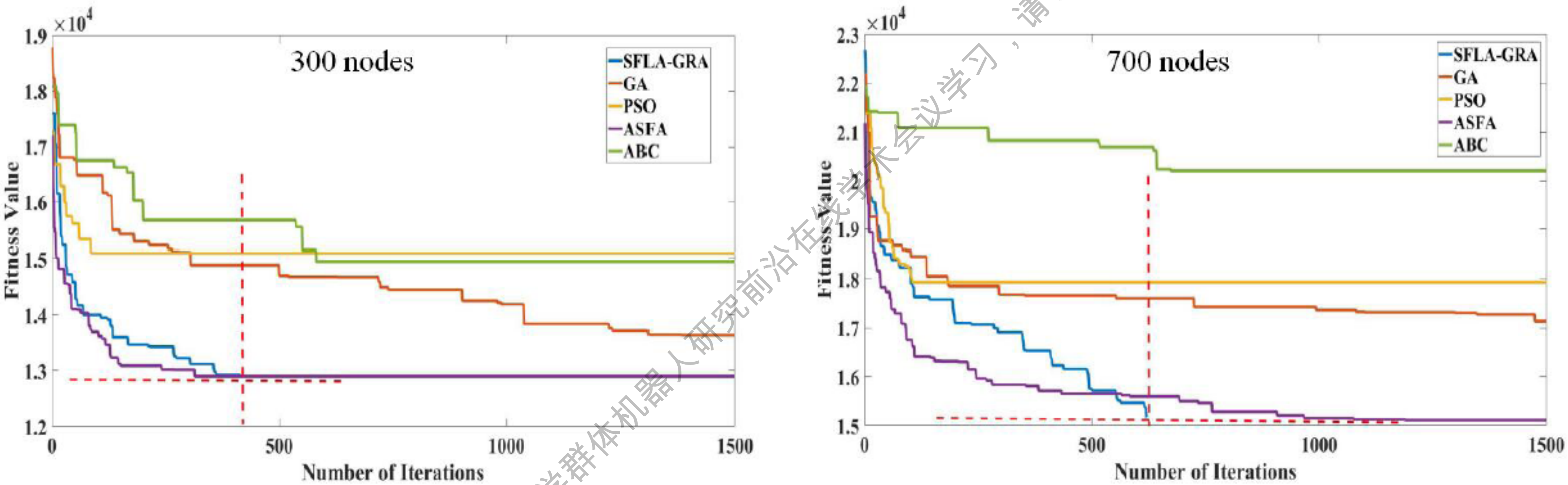


Fig. 13 SFLA-GRA shows good perfoemance among five SIs

Hardware/Partitioning Based on Brainstorm Optimization Algorithm

BrainStorm optimization (BSO) was proposed based on the collective behavior of human being, that is, the brainstorming process.

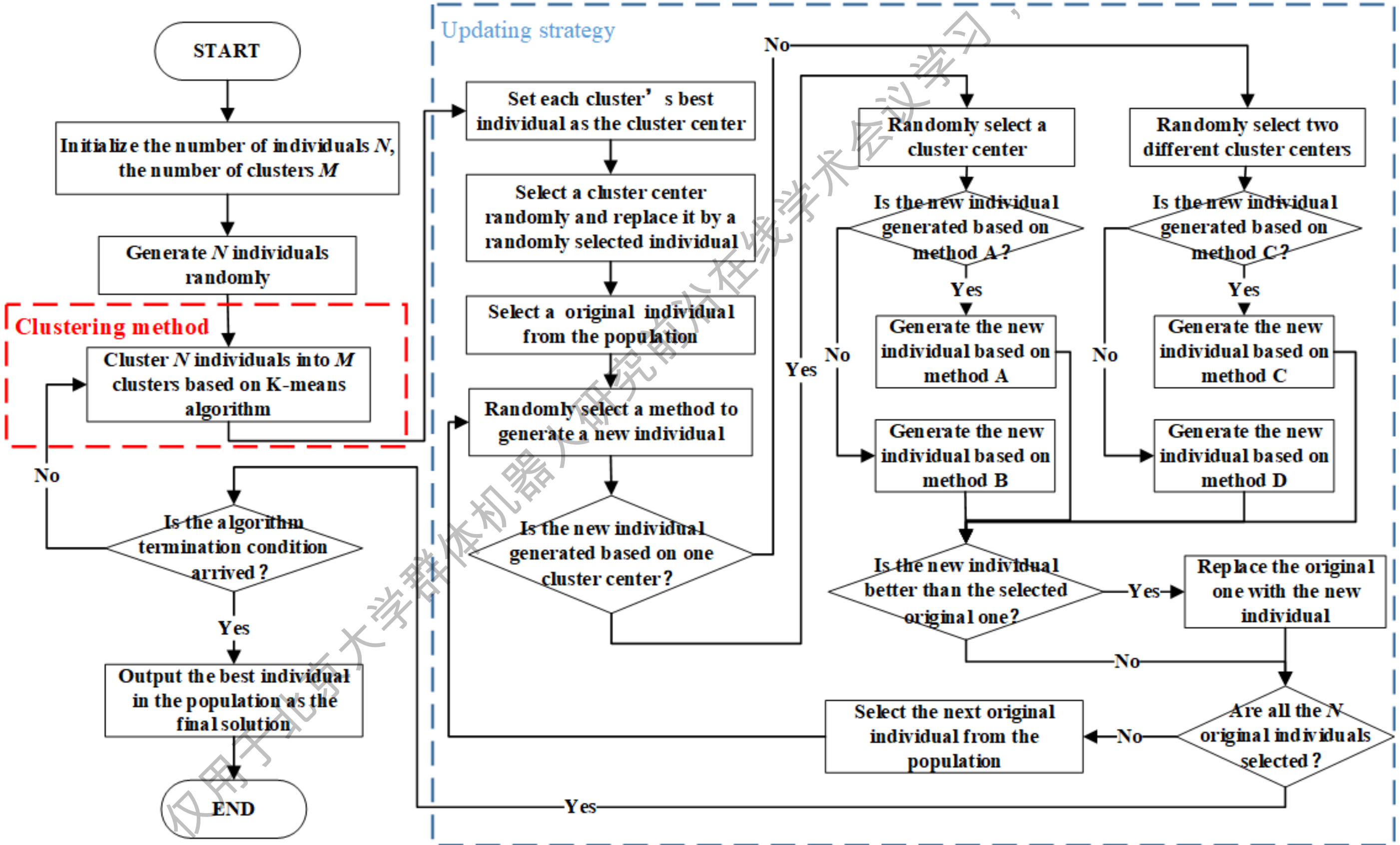


Fig. 18 The flow chart of BSO

Improvements of Brainstorm Optimization Algorithm

No.1 Original Clustering Method

The BSO uses a K-means algorithm as the clustering method and the number of individuals in each cluster may be different. When the clustering process is completed, the optimal individual in each cluster will be taken as the new cluster center and it will be utilized to guide the updating of the individuals. But this clustering method may lead individuals to gather in the same area and cause the algorithm to fall into a local optimum.

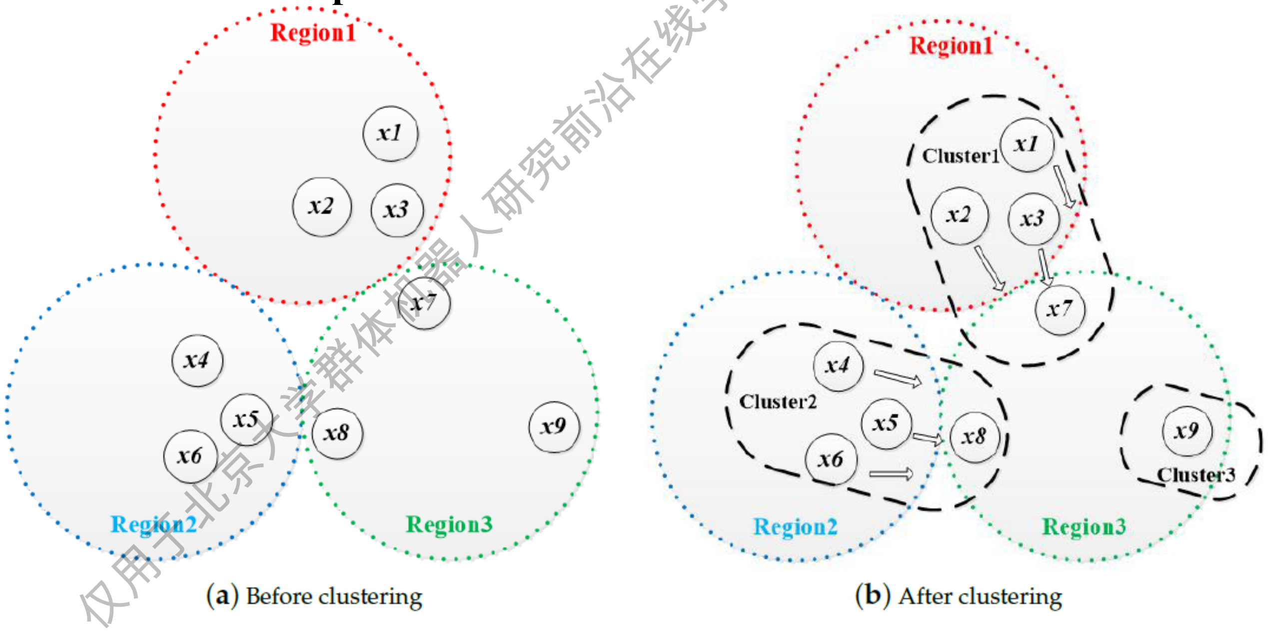


Fig. 19 An example of original clustering method

Hardware/Partitioning
Based on Improved Brainstorm Optimization Algorithm

No.1 Improved Clustering Method

Suppose that there are N individuals to be clustered into M clusters. First, setting M equidistant coordinates in the solution space which can be expressed as $C=\{c_1,c_2,\ldots,c_i,\ldots,c_M\}$, where c_i is an L-dimensional vector, and L is the dimension of the solution. The equidistant coordinates are called cluster points and c_i is the cluster point of the i-th cluster. Then all individuals are sorted according to their fitness values. Finally, clustering the sorted individuals into the cluster whose cluster point is the nearest to them.

In addition, the number of individuals in each cluster must be the same. So when the number of individuals in a cluster reaches N/M, this cluster is saturated.

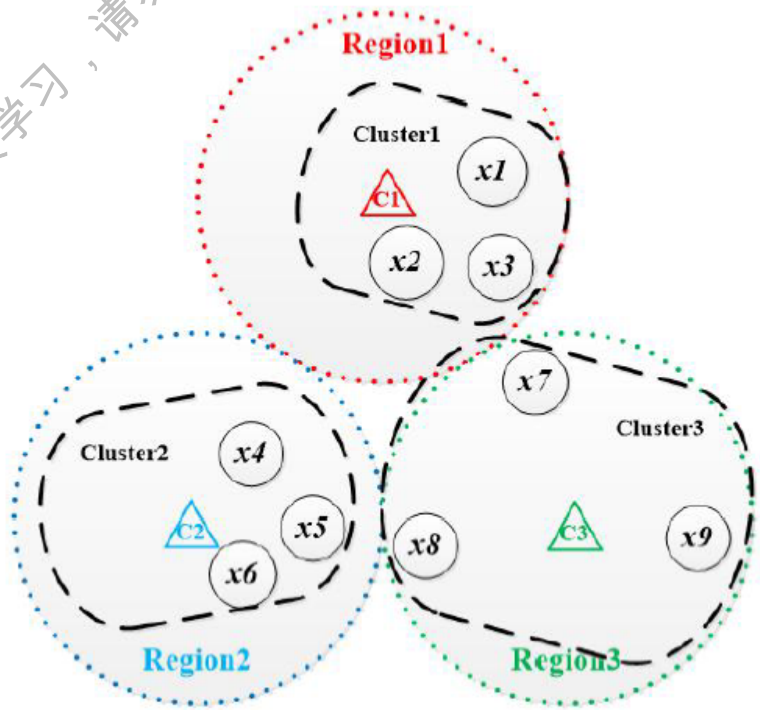


Fig. 20 Clustering result by the improved clustering method

Improvements of Brainstorm Optimization Algorithm

No.2 Original Updating Strategy

In the updating process, every individual in the population will try to search for a new individual to update itself. The new individual may be generated by adding random information to a cluster center or to the combination of two cluster centers. However, due to the solution of hardware/software partitioning problem is a vector consisting of 0 and 1, there is a great probability to generate the same or similar new individuals based on the aforementioned updating strategy. If all or most individuals in the population are replaced by the new generated individuals in each iteration, it will be difficult to keep the population diversity as the number of iterations increasing.

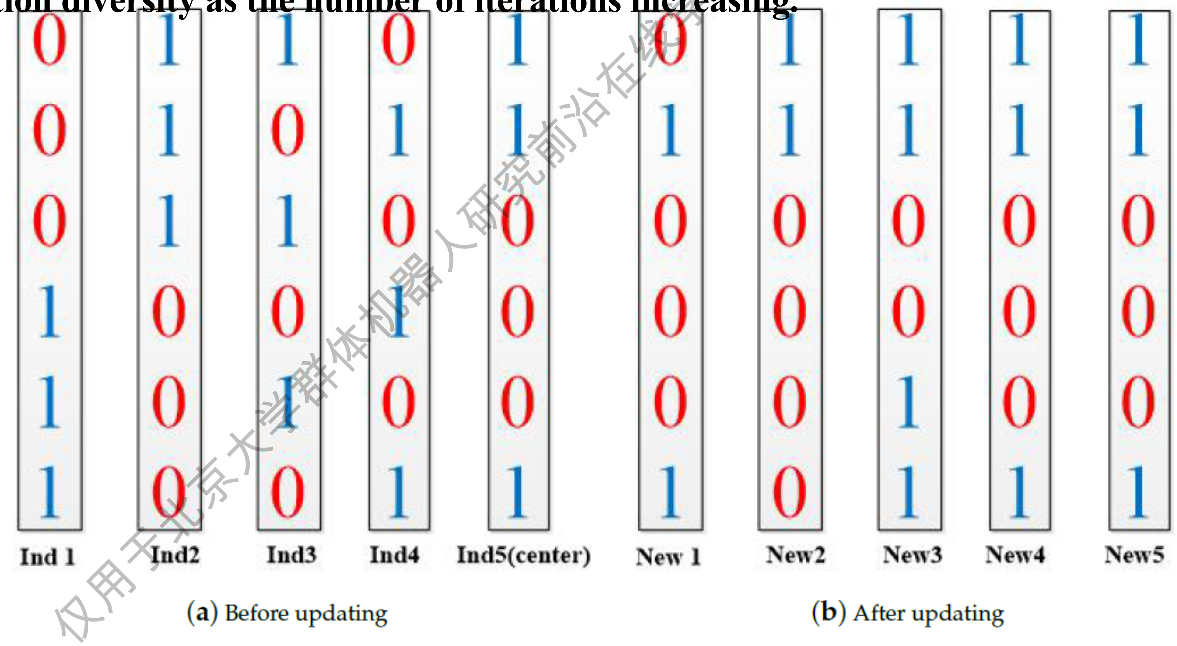


Fig. 20 An example of original updating strategy

Improvements of Brainstorm Optimization Algorithm

No.2 Improved Updating Strategy

Firstly, updating the optimal individual in the cluster. The optimal individual adds random information to generate a new individual. If the new individual is superior to the original one, replacing the original individual with the new individual. Then, randomly select an individual in the cluster and update it. The selected individual will generate a new individual by moving towards the cluster's best individual. If the new individual is superior to the original one, replacing the original individual with the new individual. Otherwise, generating an individual randomly in the solution space to replace the original one.

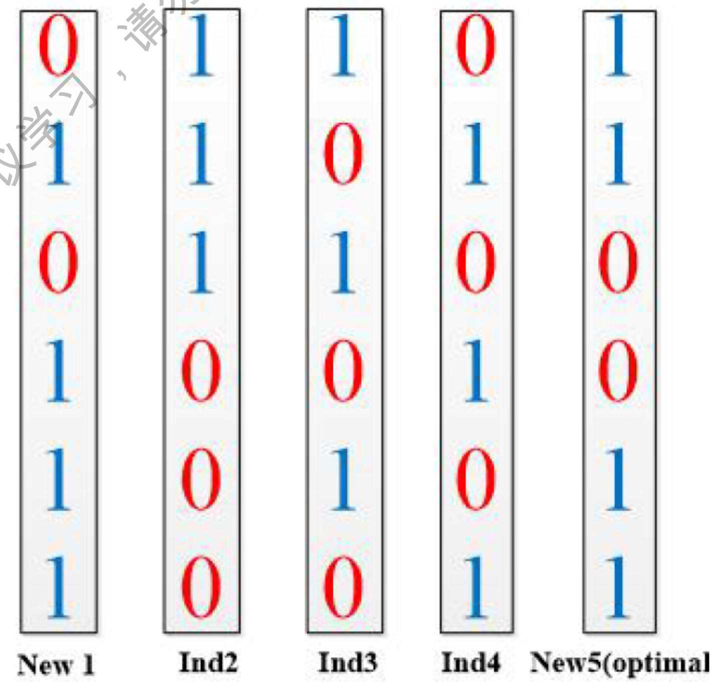


Fig. 21 Individuals updated by the new strategy

Experimental results

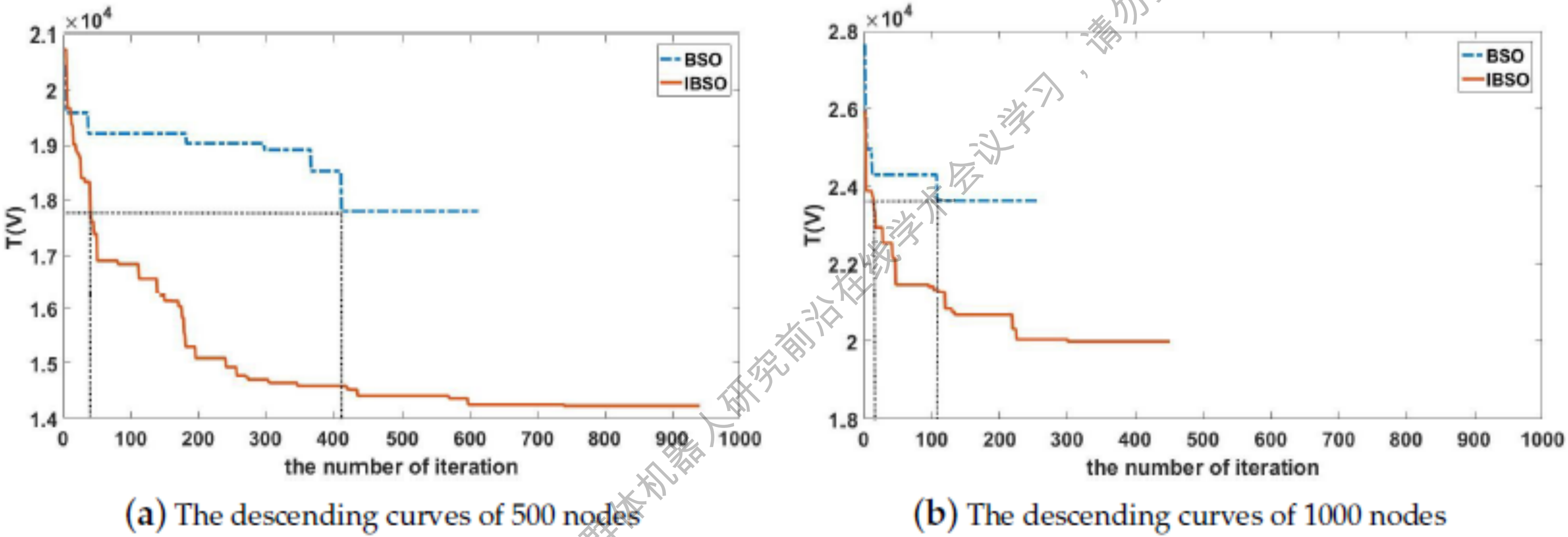


Fig. 22 Comparison between the BSO and the IBSO.

CASE TWO

Feature Selection

The Optimization Framework for Feature Selection

Given an original feature set F with N input features, there are a number of candidate subsets S with M features, that $S \subseteq F$. The objective of feature selection is to select an optimal feature subset S from F .

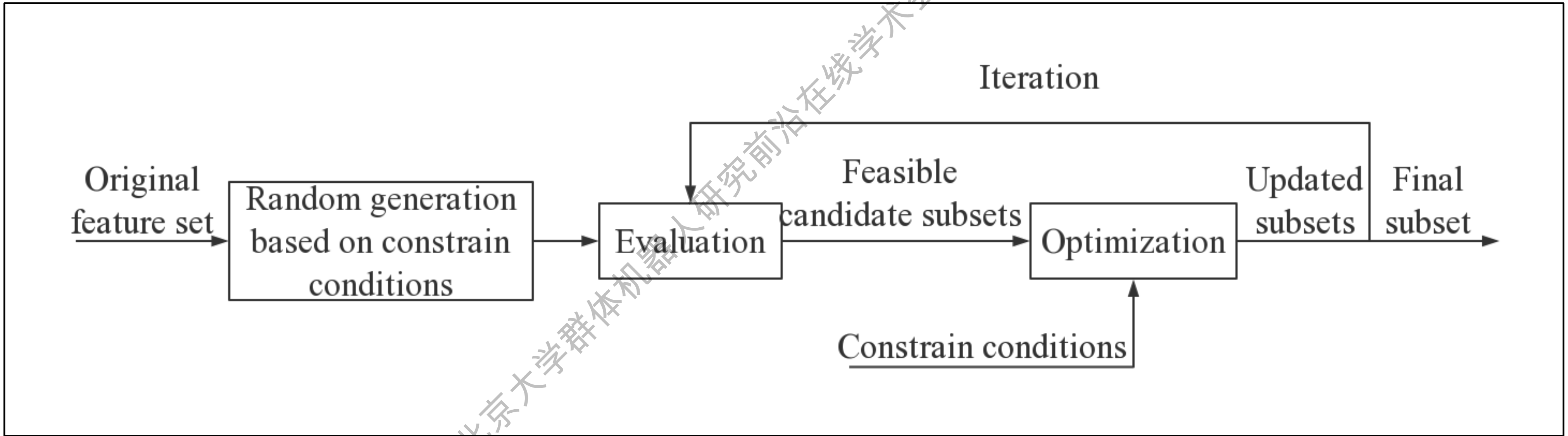


Fig. 23 Framework for feature selection

Feature Selection in Defects Detection of Glass Bottles

- Acoustic-based method is one of the most important technologies to detect the defects of objects.
- The detection process of glass bottles can be regarded as a classification task.
- A subset of features need to be selected from the et of acoustic features and then be input into the classifier.

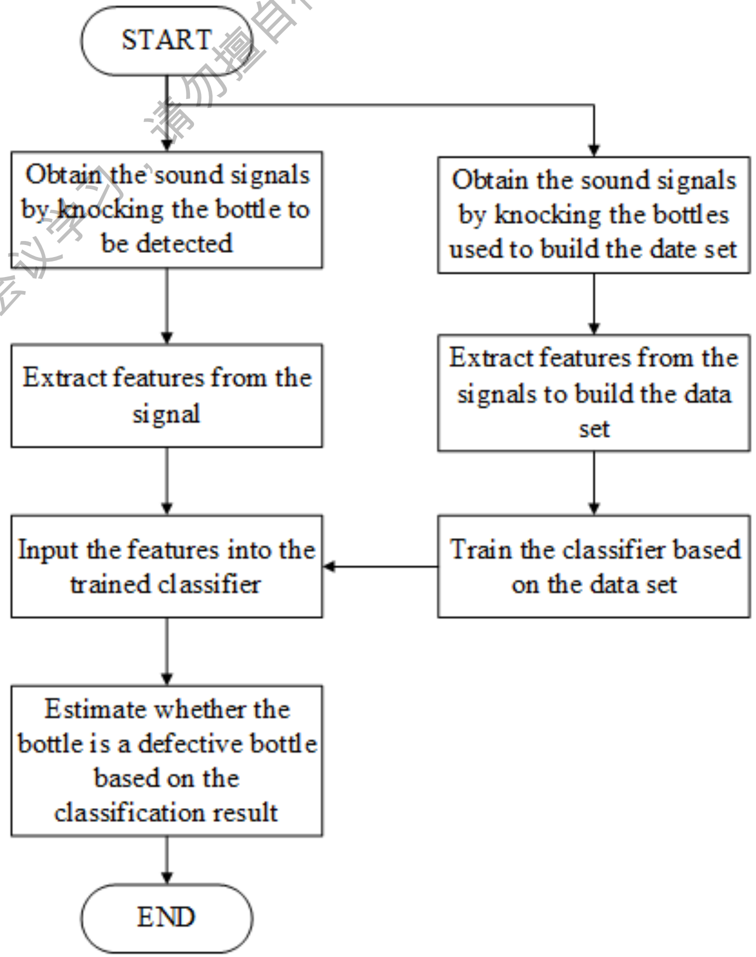


Fig. 24 The process of defect detection of a glass bottle

The Mathematical Model of Feature Selection Problem

To evaluate the selected subset of features, we proposed an **Improved minimal-Redundancy-Maximal-Relevance** (Im-RMR) method which can eliminates the irrelevant and redundant features ot the audio signal.

- During feature selection, the candidate subset is represented by a vector $S=\{s_1,s_2,\ldots,s_M\}$.
- To increase the weight of relevance between features and class label in evaluation criteria, we calculate the relevance via $D(S, L)$ .It can be expressed by (1)
- **The redundancy of features are calculated as (2)**
- **The ImRMR can be calculated by (3)**

$$D(S,L)=\sum I(s_i;L) \quad (1) \qquad R=\frac{2}{M\times(M-1)}\sum_{i<j} I(s_i;s_j) \quad (2) \qquad Eva=D(S,L)-R \quad (3)$$

The mathematical model of feature selection can be described by:

$$\begin{aligned} &\max : Eva = D(S,L)-R \\ &s.t.\;\; s_i \neq s_j (i \neq j, i\in\{1,2,\cdots,M\}, j\in\{1,2,\cdots,M\}) \end{aligned} \qquad (4)$$

The Mathematical Model of Feature Selection Problem

Representation of the feature subset:

To represent the feature subset, we adopt a real-value encoding method. Each feature subset can be represented by a M-dimensional vector consisting of 0 to $N-1$, where N is the number of features to be selected and M is the number of selected features. The elements of the vector cannot be repeated.

An example of feature selection scheme is shown as follows:

Number of feature subset	feature 1	feature 2	feature 3	feature 4	feature 5	feature 6	……
One Scheme	181	69	176	133	76	134	……

Feature Selection Based on SFLA-ImRMR Algorithm

In SFLA-ImRMR, SFLA is applied in feature selection, and ImRMR is applied to evaluate the fitness of individuals.

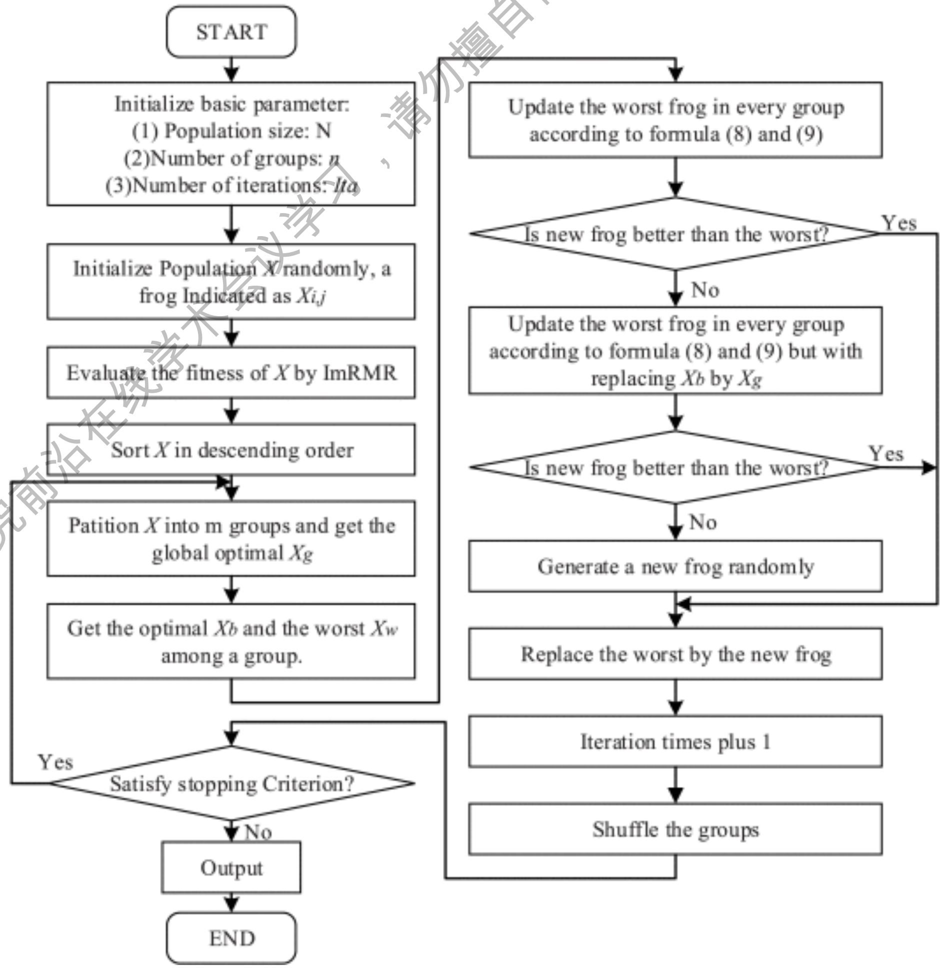
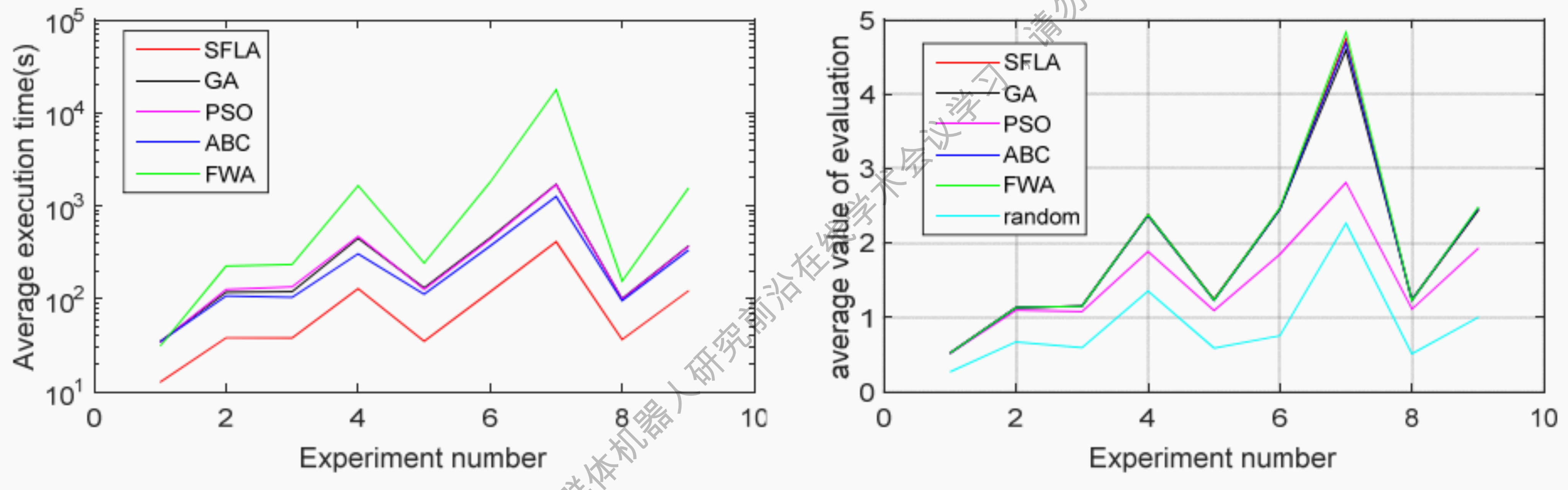


Fig. 25 The flow chart of SFLA-ImRMR algorithm.

Experimental results



(a) (b)

The execution time of SFLA is obviously shorter than the other four algorithms.
The optimal value obtained by the five algorithms is better than the random method.

Feature Selection Based on SFLA-ImRMR-BP

On the basis of SFLA-ImRMR, we propose an improve feature selection algorithm named SFLA-ImRMR-BP:

- ➢ Generate an initial solution based on the SFLA-ImRMR
- ➢ Use the wrapper approach to evaluate the fitness of the solutions
- ➢ Alternately run neural network training and feature selection

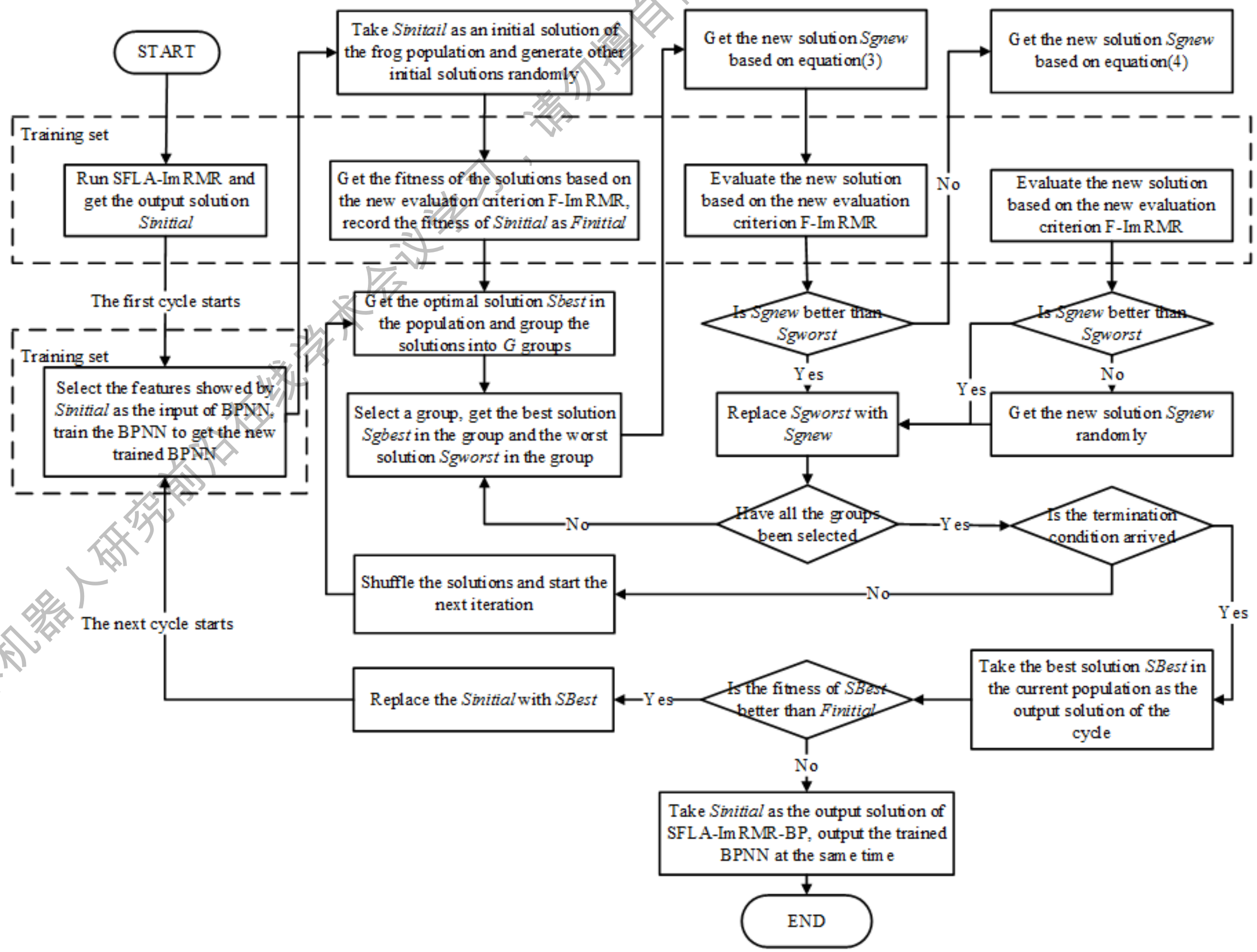


Fig. 26 The flow chart of SFLA-ImRMR-BP

Experimetal results of SFLA-ImRMR-BP

The SFLA-ImRMR-BP has the highest performance among the seven algorithms

Values of F-measure of SFLA-ImRMR and six comparison algorithms

N_M	Proposed Algorithm	SFLA -ImRMR	GA -ImRMR	PSO -ImRMR	ABC -ImRMR	FWA -ImRMR	random
50_5	**91.57%**	**91.18%**	91.02%	91.10%	90.18%	91.02%	83.70%
50_10	**91.54%**	90.91%	91.04%	90.73%	**91.08%**	90.77%	86.16%
100_10	**94.16%**	90.38%	90.52%	**91.31%**	90.56%	90.52%	85.28%
100_20	**94.19%**	90.89%	90.98%	90.54%	**91.00%**	90.86%	87.96%
200_10	**99.27%**	98.87%	**99.14%**	95.64%	98.01%	98.45%	88.64%
200_20	**99.35%**	98.41%	98.82%	92.94%	99.10%	**99.15%**	87.77%
200_40	**99.17%**	98.49%	98.09%	91.87%	97.90%	**98.79%**	90.26%
300_10	**98.74%**	94.09%	94.52%	**93.77%**	94.06%	94.37%	85.24%
300_20	**97.69%**	93.52%	93.60%	91.37%	93.71%	**93.73%**	85.71%

CASE
THREE

Local Backlight Dimming

Concepts of Local Backlight Dimming

Local backlight dimming is a promising technique which can enhance the visual quality and reduce the power consumption of LCDs. A local dimming system consists of two parts : LC panel and backlight. The backlight includes a two-dimensional array of LED backlight blocks, and each block illuminates a small region of LC panel and can be controlled independently.

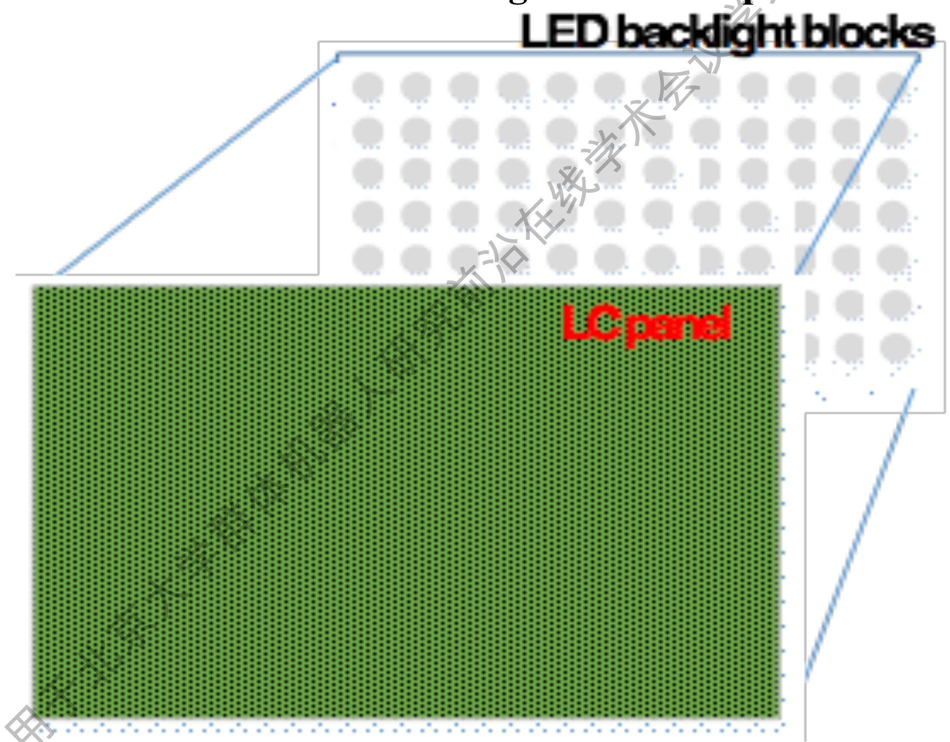


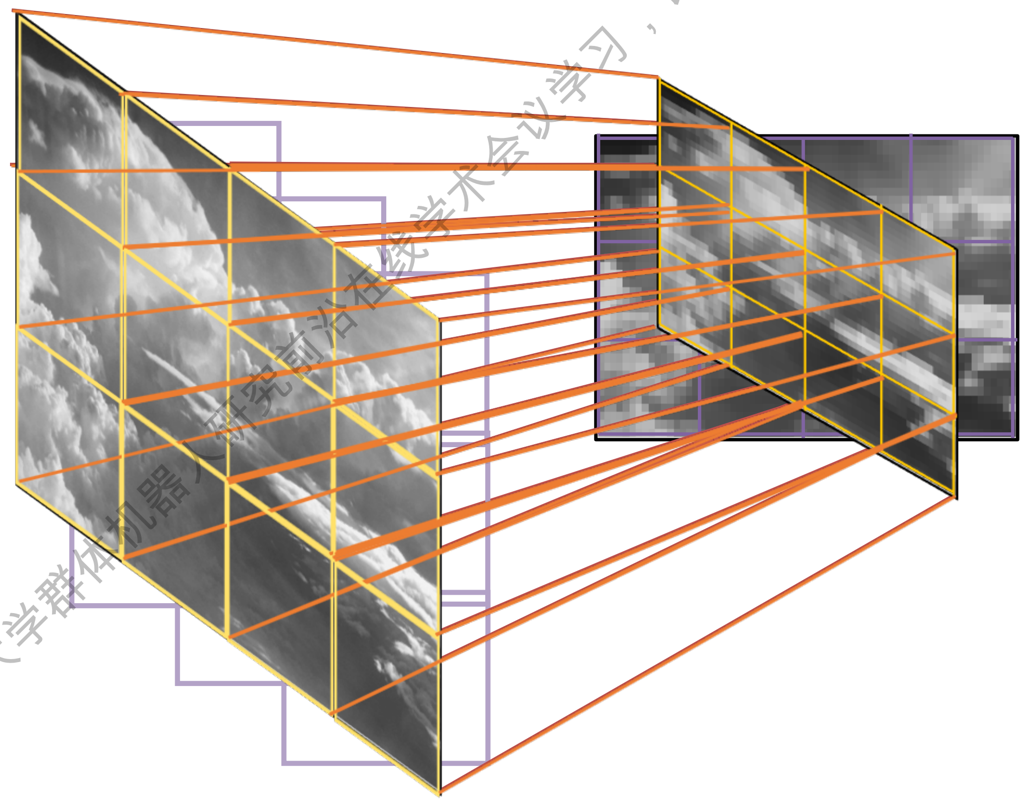
Fig. 27 The Local dimming system

Traditional Local Dimming Methods

Most of the local dimming algorithms calculate backlight luminance based on the pixel values of the image.

For example, the maximum algorithm and the average algorithm take the maximum pixel value and the average pixel value in the image region illuminated by a backlight block, respectively, as the backlight block's luminance level.

The Look-Up Table (LUT) algorithm uses a look-up table to correct the difference between the maximum algorithm and the average algorithm.



Maximum

250	255	255	255
255	250	255	255
255	200	245	255

Average

170	235	175	190
195	200	200	205
250	65	50	150

LUT

100	185	95	125
195	135	140	210
225	90	65	185

Fig. 28 Traditional local dimming methods

The Mathematical Model of Local Backlight Dimming

Visual quality and system power consumption are the two most important performance indicators

- ➢**To maximize visual quality as the optimization objective**
- ➢**Constrained by system power consumption**

Mathematical expression ：

$$\text{maximize}: \; Q(g,b)$$

$$\text{subject to}: \; pc(b) \leq pc_{LIM}$$

g : Input grayscale image

b:Backlight array

$Q(g,b)$: Displayed image visual quality

$pc(b)$: Power consumption

PC_{LIM} :Constraint value of power consumption

Optimization method: SIs

Challenge: Lack of accurate fitness evaluation function

The Mathematical Model of Local Backlight Dimming

PSNR can indicate the degree of image distortion stemming from backlight. PSNR is defined as the following:

$$\begin{cases} PSNR = 10 \cdot \log_{10}(\frac{255^2}{MSE}) \\ MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (Y'_{i,j} - Y_{i,j}) \end{cases}$$

Where H and W are the height and width of the input image, $Y_{i,j}$ and $Y'_{i,j}$ are the luminance of (i, j)th pixel before and after local dimming, respectively.

The mathematical model can be expressed by:

$$\max : f(x) = PSNR$$
$$s.t.\ PC \leq P_{limit}$$

Disadvantages of SFLA for Local Backlight Dimming

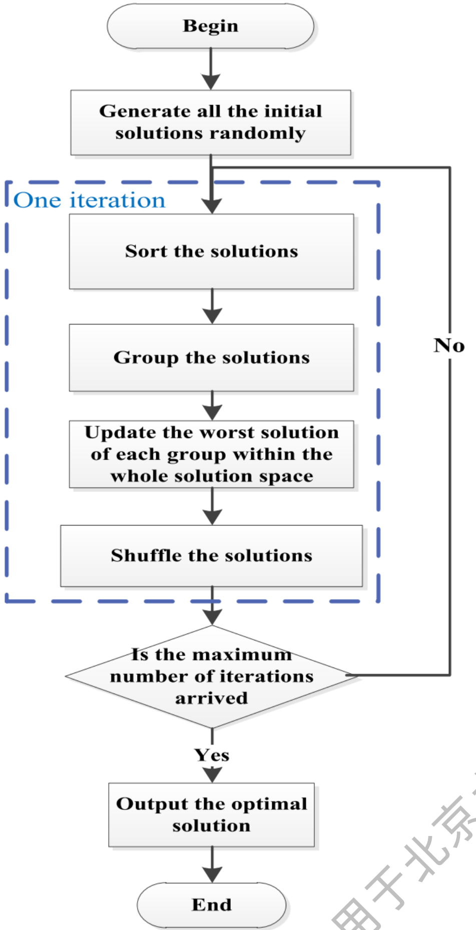


Fig. 29 The flow chart of SFLA

Disadvantages of SFLA：

- **The SFLA randomly generates a set of initial solutions, and then updates the initial solutions to get better solutions. When the algorithm is running in a large solution space and the initial solutions' quality is poor, it is difficult to obtain high quality solutions.In this case, the algorithm execution time may be very long and the quality of the final output solution may be poor.**
- **The range of backlight luminance value is [0, 255], so the search interval of solutions is usually set to [0, 255]. In fact, there is a certain correlation between the pixels' gray levels and the backlight luminance. Serious distortion may occur in some image regions after local dimming if the search interval is not limited according to this correlation.**

Comparision of SFLA and ISFLA

SFLA is applied to solve local dimming algorithm, and we improve SFLA to propose ISFLA.

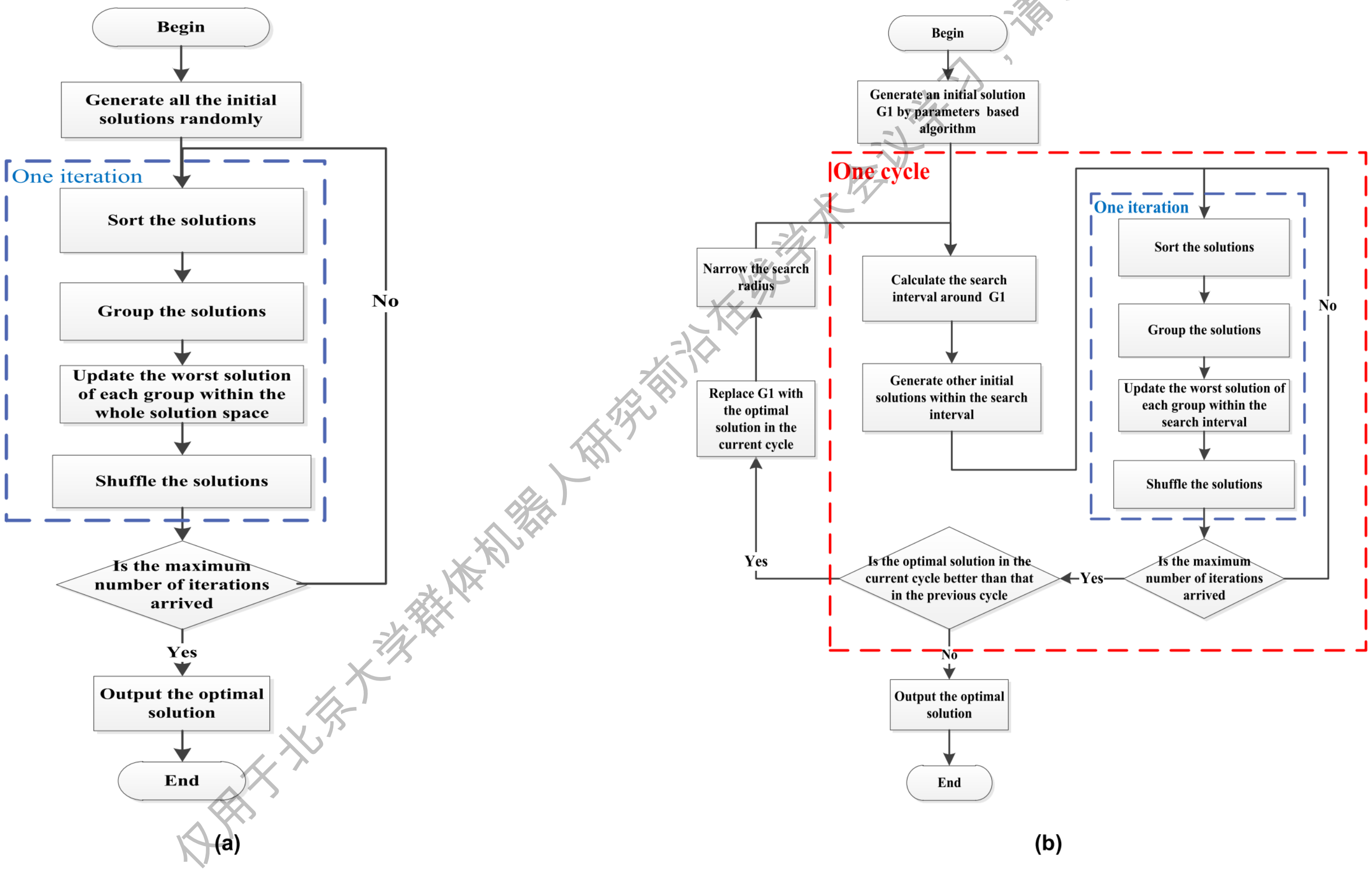


Fig. 30 The flow charts of SFLA and ISFLA. (a) SFLA. (b) ISFLA.

Steps of ISFLA

On the basis of SFLA, cycle optimization is introduced in ISFLA.The specific steps of one cycle of ISFLA are described as follows:

1) Generate the first initial solution. In the first cycle, the first initial solution should be generated by an existing algorithm based on image parameters such as the maximum algorithm or the average algorithm. In other cycles, the first initial solution is generated based on the method proposed in step (5).

2) Obtain the search interval of this cycle. The first initial solution will be set as the center of the search interval, the lower and upper bounds are obtained by subtracting the interval radius from the center and adding the interval radius to the center, respectively.

3) Generate other initial solutions of this cycle. Other initial solutions will be generated randomly in the search interval determined in step (2).

4) Start the iterative updating of this cycle. The search for new solutions should be done within the search interval obtained in step (2).

5) At the end of this cycle, the optimal solution of this cycle is taken as the first initial solution of the next cycle.

6)Start the next cycle.

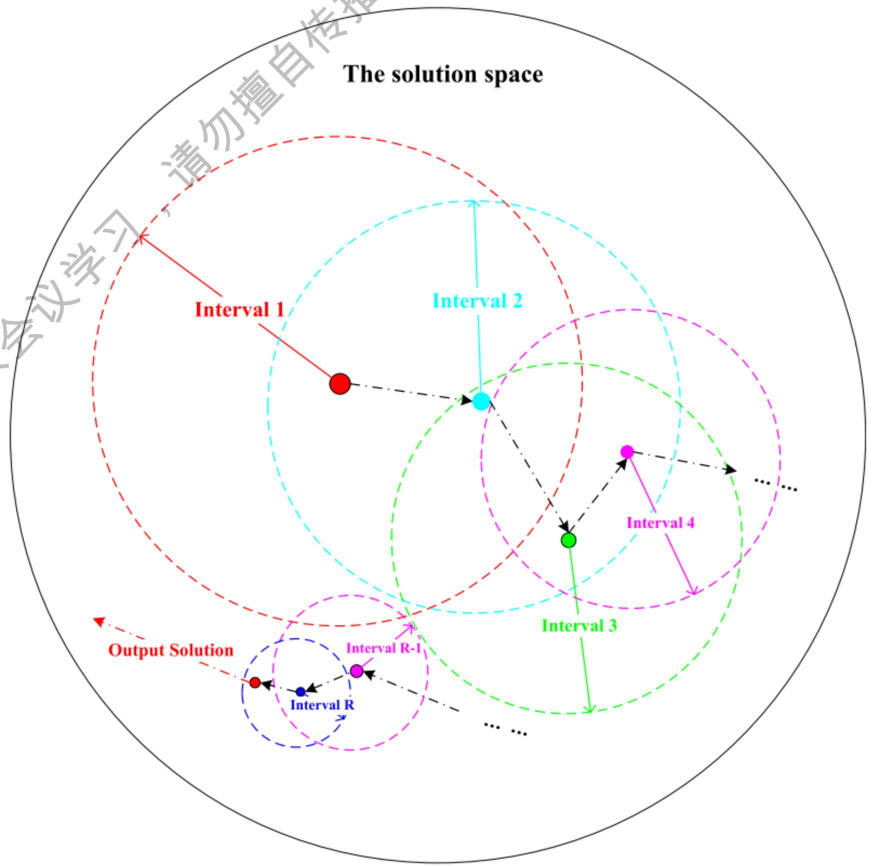


Fig. 31 The diagrammatic sketch of ISFLA algorithm process

Experimental results

The traditional method ISFLA

Fig. 32 The simulation results of local dimming

The New Mathematical Model of Local Backlight Dimming

Because SSIM index usually has a better performance than PSNR in terms of image quality evaluation. We proposed a new mathematical model in which the mean structural similarity (MSSIM) index is applied to evaluate the image quality.

The new mathematical model of local backlight dimming is built as follows:

$$\begin{cases} \min: \ f(R) = \frac{1}{\mathrm{MSSIM}(X, Y(R))} \\ \text{s.t.} \ \frac{\frac{1}{K}\sum_{i=1}^{K} r_i}{r_{full}} \leq PC_{limit} \end{cases}$$

Fireworks Algorithm(FWA)

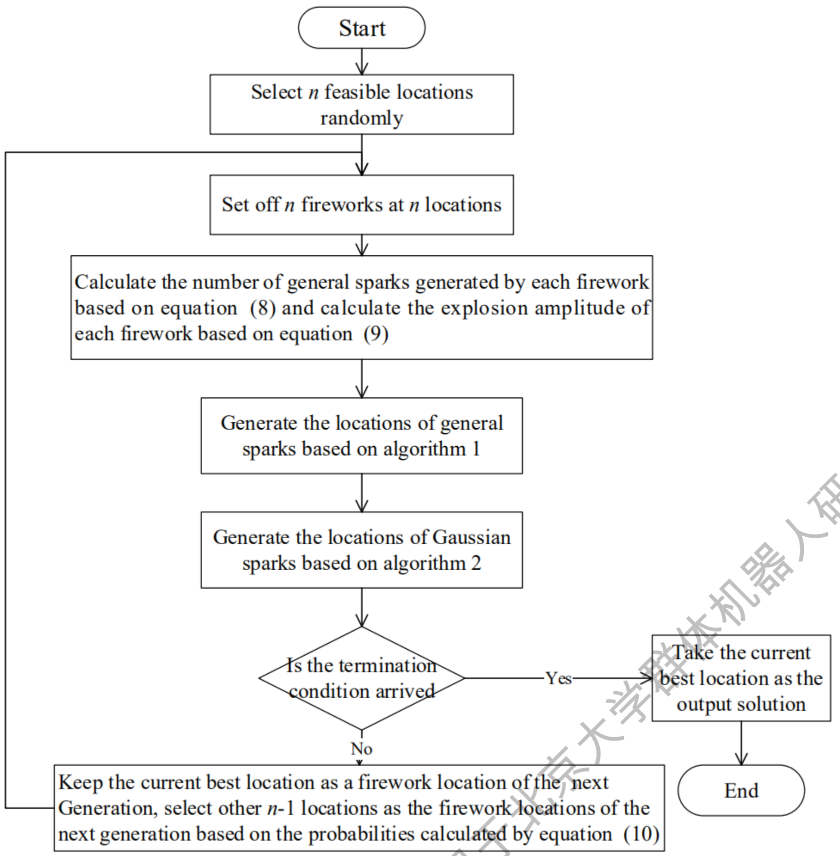


Fig. 33 The flow chart of FWA.

Fireworks Algorithm：

- FWA is an EC algorithm which was developed by simulating the explosion process of fireworks.
- When FWA is used to solve optimization problems, the firework and spark locations correspond to the solutions of the problem.

The Search Process of FWA:

- New spark locations are generated by random searching around the locations of fireworks.
- For different firework locations, the search step size and the number of generated spark locations are different. If a firework location has better quality, there will be more spark locations generated around it, and its search step size will be smaller.
- High quality spark locations appear more easily near high quality firework locations, this search strategy is conducive to improve the search ability of the algorithm.

Guided FWA(GFWA)

FWA is applied to solve the new model, and we improved the performance of FWA by proposing a guided FWA (GFWA).

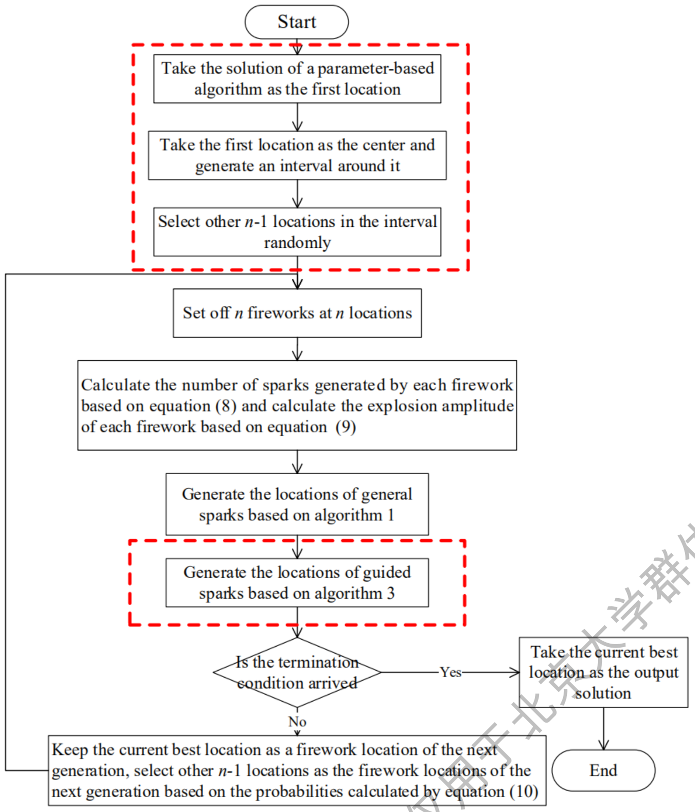


Fig. 34 The flow chart of GFWA.

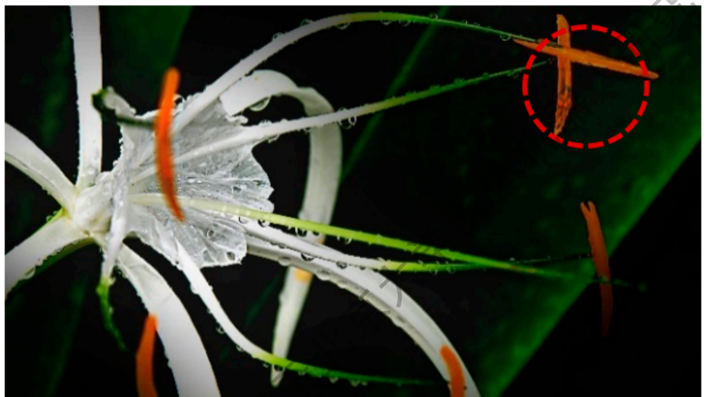
There are three improvements in GFWA:

- **The solution of the parameter-based algorithm is taken as the first firework location,and other firework locations are generated within an interval which is centered on the first location.**
- **In the search process of GFWA, the strategy of generating guided sparks is applied. After the general spark locations are generated, the firework locations and the general spark locations are mixed together.**
- **By the guidance of R_{best} , if a bad location can't find a better guided spark location to replace itself, it will be updated by some strategies.**

Experimental results

(a)

(b)



(c)

(d)

The simulation results of local dimming. Image. (a) GFWA, (b) LUT, (c) FWA, (d) ISFLA

Q&A

THANKS

zhangtao@tju.edu.cn