

The Application of Improved Bacteria Foraging Algorithm to the Optimization of Aviation Equipment Maintenance Scheduling

Jia CUI

Abstract: Taking the aviation equipment scheduled maintenance as a prototype, this paper improves a bionic global random search algorithm - bacteria foraging optimization algorithm to solve the task-scheduling problem. Inspired by gene mutation, the activity of bacteria is dynamically adjusted to make good bacteria more capable of action. In addition, a bacterial quorum sensing mechanism is established, which allows bacteria to guide their swimming routes by using their peer experience and enhance their global search capability. Its application to the engineering practice can optimize the scheduling of the maintenance process. It is of great application value in increasing the aviation equipment maintenance efficiency and the level of command automation. In addition, it can improve the resource utilization ratio to reduce the maintenance support cost.

Keywords: bacteria foraging optimization algorithm; job scheduling; quorum sensing mechanism; scheduled maintenance; variable step size

1 INTRODUCTION

Bacteria foraging optimization (BFO) algorithm was proposed by Passino based on a bionic random search algorithm in 2002. The biological foundation is that E.coli bacteria have some intelligent behaviours in their foraging process. There is a control system in E.coli bacteria to guide their behaviours [1, 2], including chemotaxis, reproduction and elimination-dispersal in the foraging process. In addition, it can evaluate the effect of state change made by bacteria each time to provide information for their next movement. Under the control of this system, E.coli bacteria will gradually move towards the food source. In our BFO model, the advantages and disadvantages of the search algorithm are judged according to the nutrient distribution function in the searching process, where the solution to the problem corresponding to the state of bacteria in the search space will be optimized. In other words, the fitness value of the function will be optimized.

The aviation equipment scheduled maintenance involves several workshops, jobs and processes. However, our optimization algorithm can realize the effective organization, precise and efficient scheduling of the regular maintenance work to achieve the optimal combination and reasonable distribution of the various supporting resources, and reduce the maintenance support cost. In addition, it can be applied in the virtual-reality (VR) optimal scheduling system for the aviation equipment scheduled maintenance after the BFO algorithm is improved.

2 THE IMPROVEMENT OF THE ORIGINAL BFO ALGORITHM

2.1 Induction Mechanism in Bacteria Foraging Behaviour

For the search in the solution space, the original BFO algorithm adopts the cell-to-cell sensing mechanism. It calculates the cell-to-cell sensing value [3] according to Eq. (1) when a bacterium swims to a new position each time:

$$J_{cc}(\theta^i(j, k, l), \theta(j, k, l)) = \sum_{t=1}^S J_{cc}^t(\theta^i, \theta) = \sum_{t=1}^S \left[-d_{\text{attract}} \exp\left(-\omega_{\text{attract}} \sum_{m=1}^P (\theta_m^i - \theta_m^t)^2\right) \right] + \sum_{t=1}^S \left[h_{\text{repellant}} \exp\left(-\omega_{\text{repellant}} \sum_{m=1}^P (\theta_m^i - \theta_m^t)^2\right) \right] \quad (1)$$

Where: $J_{cc}(\theta^i(j, k, l), \theta(j, k, l))$ represents the sum of the cell-to-cell sensing values when the i^{th} cell is undergoing the j^{th} chemotaxis operation, the k^{th} reproduction and the l^{th} elimination-dispersal operation. $J_{cc}^t(\theta^i, \theta)$ represents the value of the quorum sensing between the i^{th} cell and the t^{th} cell. d_{attract} and ω_{attract} separately denote the number and the release rate of the attractants. $h_{\text{repellant}}$ and $\omega_{\text{repellant}}$ are separately the number and the release rate of the repellants. S stands for the bacterial population size, p represents the dimension of the bacteria search environment and θ_m^i denotes the m^{th} dimension of the position, where the i^{th} cell is located.

$$\begin{cases} \overline{J^i}(j+1, k, l) = J^i(j+1, k, l) + J_{cc}(\theta^i(j+1, k, l), \theta(j+1, k, l)) \\ J^i(j+1, k, l) = \overline{J^i}(j+1, k, l) \end{cases} \quad (2)$$

According the Eq. (2), add the cell-to-cell sensing value of bacteria that had completed $(j+1)^{\text{th}}$ chemotaxis to the fitness value.

Such a cell-to-cell sensing mechanism can help maintain the population diversity and increase the possibility of escaping from local optima by bacterial swarm. However, there is a possibility that the optimization process will be delayed. When numerous bacteria gather around the global optimal location, the concentration of exclusion factor will increase. At this time, the bacteria in the best position may be expelled from the optimum area, which will cause a decrease in accuracy [3].

Inspired by the quorum sensing mechanism in the particle swarm optimization (PSO) algorithm [4-6], holding that each particle can make self-evaluation, learn

from the best particle and move towards the historical best position in the swarm, the BFO algorithm can be improved by performing the quasi-PSO based cell-to-swarm sensing operation (or the cell-to-swarm sensing value) after the chemotaxis operation ends each time: By sensing the surrounding environment, an individual cell would try to search if there is a cell that occupies the best position in the bacterial swarm. If yes, that position will be memorized. Otherwise, the current position of the cell will be memorized. After that, the cell will always swim towards the memorized position according to Eq. (3) every time when it swims to a new position in the chemotaxis operation.

If $J^i(j+1, k, l) > J_{best}(j, k, l)$ (3)
 $\theta_{cc}^i(j+1, k, l) = \theta^i(j+1, k, l) + C_{cc} \times (\theta^b(j, k, l) - \theta^i(j, k, l))$

Where: $\theta_{cc}^i(j+1, k, l)$ represents the updated position of the cell i after the quorum sensing. $\theta^b(j, k, l)$ and $J_{best}(j, k, l)$ separately denote the currently best position and the fitness value of the cell within the whole swarm. C_{cc} is the attractant deciding the step size [7] of the cell when it swims to the historically best position in the swarm. After the introduction of the quasi-PSO operation to make the improvement, the velocity and the position of the cell i can be updated according to Eqs. (4) and (5):

$$V_{id}^{new} = \omega \cdot V_{id}^{old} + C_1 \cdot \varphi_1 \cdot (\theta_{g_best} - \theta_d^{old}(i, j+1, k)) \quad (4)$$

$$\theta_d^{new}(i, j+1, k) = \theta_d^{old}(i, j+1, k) + V_{id}^{new} \quad (5)$$

Where: ω is the inertia weight that enables the cell to maintain the movement inertia and expand the search space to explore a new region. C_1 is the learning factor, or the acceleration constant. φ_1 is the pseudo-random numbers distributed uniformly within the interval of [0,1] and \vec{V}_i represents the velocity vector of the cell i .

After the improvement, our new quorum sensing mechanism allows a cell to guide its own dispersal route based on the experiences of the other cells. Such a belief attraction mechanism has accelerated the global searching speed in the solution space and at the same time, it will facilitate an individual cell to escape from the local optima to avoid the possibility that a cell might not be covered in the global optimal area.

2.2 Variable Step Size in Bacterial Chemotaxis Operation

In the original BFO algorithm, the forward chemotactic movement of a cell can be calculated according to Eq. (6) [8]:

$$\theta(i, j+1, k) = \theta(i, j, k) + C(i)\Phi(j) \quad (6)$$

Where: $F(j) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$, ($\Delta(i)$ is the direction vector within the interval of [-1, 1]).

Genetic mutation might occur in the life cycle of a cell. Excellent bacteria with stronger chemotactic ability might

have a bigger chemotaxis step in the elimination-dispersal process to speed up the bacterial foraging process. In order to further accelerate the searching speed, improve the global searching ability at the initial searching stage and the local searching ability in the later period, the BFO algorithm can be further improved by setting the step length into a dynamic adjustment value instead of a fixed value [7] in the chemotaxis operation according to Eq. (7):

$$C(k, l) = \frac{L_{red}}{n^{k+l-1}} \quad (7)$$

Where: $C(k, l)$ represents the step length for the chemotaxis operation when the cell is undergoing the k^{th} reproduction in the l^{th} elimination-dispersal event. L_{red} is the initial chemotaxis step length and n denotes the parameter used to control the step-length gradient descent.

2.3 Design of an Improved Fast Bacterial Swarming Algorithm

On the basis of adjusting the step length and setting up the induction mechanism, the process of the algorithm is improved. Before the chemotaxis operation, the bacteria produce variation and change the step length. Then, the individual induction and the quorum induction will be carried out to the optimal foraging area. In addition, bacteria are allowed to mutate and adjust step lengths during migration. The improved algorithm flow chart is provided in Fig. 1.

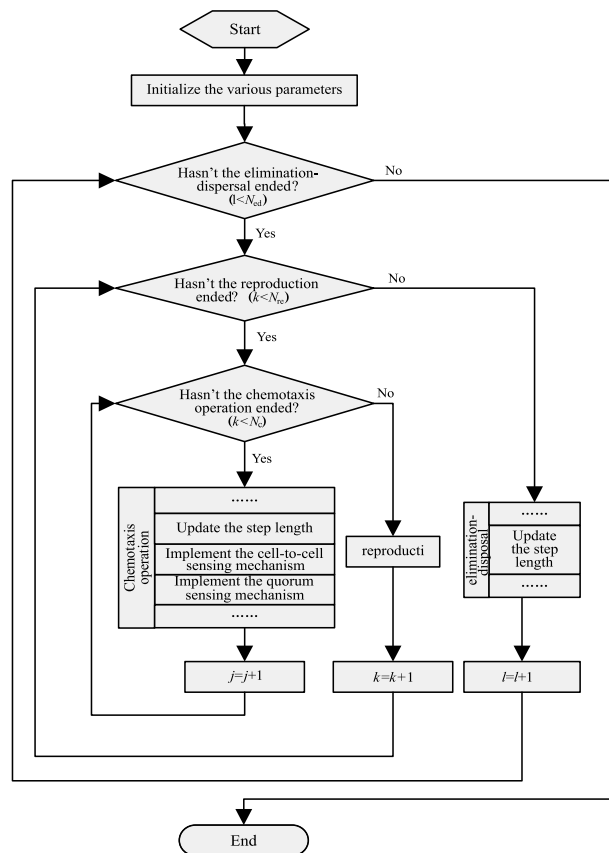


Figure 1 The improved algorithm flow chart

At the initial optimization stage, or when the value of $k + l$ is small, the increase in $C(k, l)$ means that the cell has

better movement ability in the solution space, and less searching time will be spent by the cell in a local scope. But in the latter period, the value of $k + l$ will gradually go up with a decrease in $C(k, l)$. It indicates that the cell has stronger local searching ability when it swims nearer to globally optimal point, proving that the algorithm will finally arrive at the globally optimal point. When it is mapped onto the bacteria foraging behaviours, it is equivalent to the improvement of bacteria foraging ability at a quicker speed. From the perspective of bacterial swarm, it can enhance the elimination-dispersal ability of the whole bacterial swarm as the genes of the excellent bacteria have been preserved.

2.4 Optimization of the Nonlinear Model
2.4.1 Typical Nonlinear Model

Choose Shaffer function as the typical nonlinear model for the simulation and it can be described as below:

$$z = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{\left[1 + 0.001 \cdot (x_1^2 + x_2^2)\right]^2} \tag{8}$$

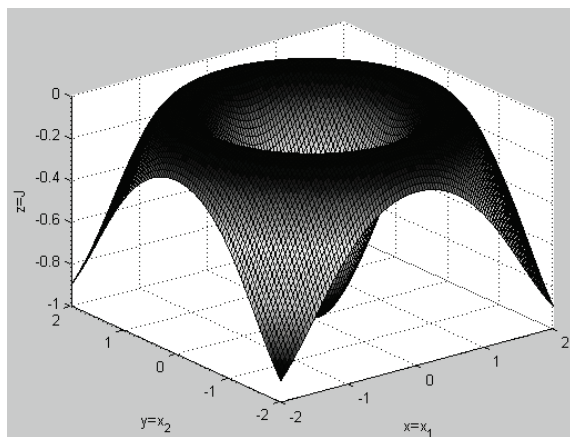


Figure 2 The topographic map of energy concentration (Valley=food, Peak=hazardous substances)

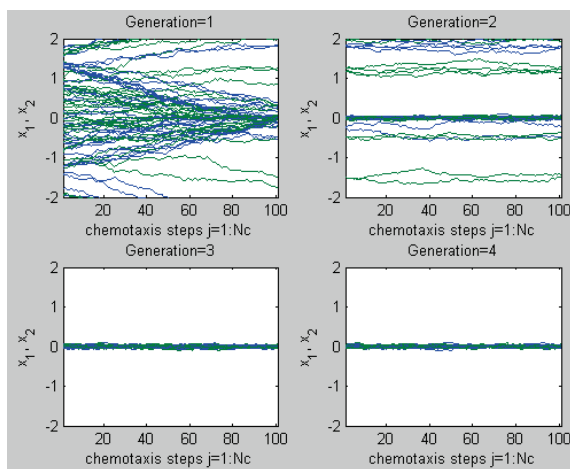


Figure 3 The bacteria dispersal trajectory from the first to the fourth generation in the first elimination-dispersal event

Where: $x_1, x_2 \in [-2, 2]$. Choose z as the fitness function and the population size $S = 40$, the number of chemotactic behaviours $N_c = 100$ and the maximum

number of chemotaxis steps $N_s = 4$. Also set the number of generations for reproduction $N_{re} = 4$, the number of elimination-dispersal events $N_{ed} = 2$ and the elimination-dispersal probability $P_{ed} = 0.25$. The simulation result is shown in Figs. 2 - 4.

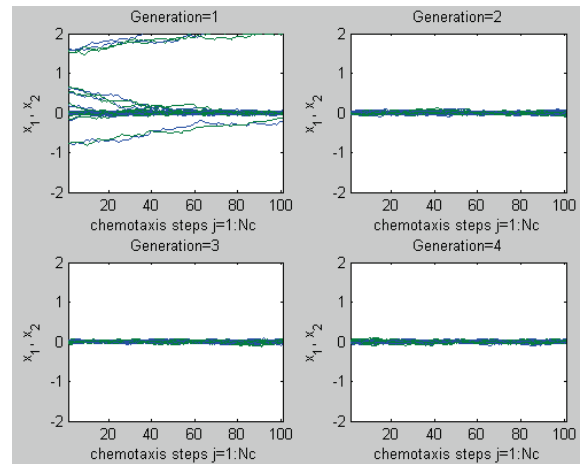


Figure 4 The bacteria dispersal trajectory from the first to the fourth generation when the number of elimination-dispersal events is 2

2.4.2 Analysis on the Simulation Result

The simulation result reveals that the first generation of bacteria are dispersed randomly in the to-be-optimized region, where they gather in swarms to perform the chemotaxis behaviours. After the screening, the bacteria reproduce. The second generation of bacteria will repeat the behaviours of the bacteria in the first generation by constantly swarming toward the food. Till the fourth generation, they have clumped together within a small range. In the second elimination-dispersal event, the bacteria start from the elimination-dispersal operation according to the elimination-dispersal probability based on the previous four generations and then continue their searching behaviours. Basically they have gathered in the optimal region in the second generation, and continued to converge in this region in the next two generations till they have converged into a very small range in the fourth generation. All of these have proved the effectiveness of this algorithm. As the bacteria have converged into the optimal region in the second elimination-dispersal event, it proves that this algorithm is very simple with a quick convergence speed.

1) Test of iteration numbers

Make a comparison on the required number of iterations when the precision is 0.00001. The results are provided in Tab. 1.

Table 1 Comparison of iteration number

| Test function | Optimization methods | The minimum number of iterations | The average number of iterations | The maximum number of iterations |
|------------------|----------------------|----------------------------------|----------------------------------|----------------------------------|
| Shaffer function | PSO | 8 | 33.46 | 143 |
| | BFO | 4 | 32.72 | 150 |

2) Convergence test

Set the maximum number of iterations as 100 and run every algorithm for 100 times to calculate the average optimal solution. The result is provided in Tab. 2.

Table 2 Comparison of the average optimal solutions

| Test function | Optimal solution | PSO | BFO | Improved BFO |
|------------------|------------------|--------|--------|--------------|
| Shaffer function | -1 | -0.986 | -0.983 | -0.995 |

The above data analysis reveals that the BFO algorithm works very efficiently in solving the model optimization problem. Also the improved algorithm can significantly increase the searching ability to avoid effectively the trap of local optimum.

3 THE APPLICATION OF BFO ALGORITHM TO SCHEDULING OPTIMIZATION

3.1 Description of Scheduling Problem

Generally, for a workshop with n work pieces (corresponding to the to-be-repaired aircraft components in an aircraft repair factory) and m machines (corresponding to the maintenance support equipment used in the workshop in an aircraft repair factory), the determination on the start times s_{ij} of every process and the processing time t_{ij} to achieve the total flow time minimization f_{max} in a feasible schedule constitutes the maintenance scheduling problem. That is:

$$f_{max}^* = \min(f_{max}) = \min \{ \max(s_{ij} + t_{ij}) : G_i \in G, M_j \in M \} \tag{9}$$

If there is a definite job sequence matrix M_j^* that enables the time function for the whole maintenance process to meet Eq. (9) and conforms to the machine sequence matrix G_M , then M_j^* can be considered as the optimal solution [9] of the target function used to solve the job shop scheduling problem (JSSP).

Now in a common maintenance support system, every equipment can be maintained in a specific maintenance organization (such as the maintenance unit or office with a fixed location). Every maintenance unit can support the maintenance of equipment from several units. However, in normal cases, the aircraft scheduled maintenance is always performed in the place where the maintenance team is located, which usually is the dock or the landing field of the repair plant. Therefore, the distance between the support resource, the failure equipment and the maintenance team will not be taken into account.

Use the operator of the bacteria foraging algorithm described in Eq. (10) to express the processing time [10] consumed by a workpiece in the processing sequence of a certain machine:

$$T_i = g(J(i, j, k, l)) \tag{10}$$

Where, $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l))$. It represents the job sequence of the operator used in the bacteria foraging algorithm. Use the bacteria foraging algorithm to search the sequence and then obtain M_j^* . This is the application of the swarm-based bacteria foraging algorithm in the job shop scheduling system.

The general procedure for the application of the improved BFO algorithm to solving the scheduling optimization problem is as below:

- 1) Problem representation;
- 2) Implement the decoding strategy;
- 3) Design an evaluation function and create the initial solution population;
- 4) Use the quorum sensing mechanism for global searching.

3.2 Problem Representation

With the adoption of the operation-based representation strategy, every feasible solution can be expressed with $n \times m$ numbers on behalf of the operations to constitute an array of operations conforming to the process constraints. Every number in the array represents the part number of the to-be-repaired component. The j^{th} emergence of the to-be-repaired component i represents the j^{th} operation or the j^{th} process of the to-be-repaired component i ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$). As process constraints have already been taken into account in the representation, then a feasible solution can be obtained during the displacement transformation of the numbers. Tab. 3 gives the process constraints on typical 3×3 scheduling problems, and Fig. 5 gives an example of representation [11].

Table 3 Process constraints on 3×3 job shop scheduling problems

| Part No. of the to-be-repaired components | No. 1 process | | No. 2 process | | No. 3 process | |
|---|---------------|-------------|---------------|-------------|---------------|-------------|
| | Machine No. | Repair time | Machine No. | Repair time | Machine No. | Repair time |
| 1 | 2 | 3 | 3 | 7 | 1 | 6 |
| 2 | 1 | 6 | 3 | 9 | 2 | 8 |
| 3 | 3 | 4 | 1 | 3 | 2 | 9 |

Under the process constraints provided in Tab. 3, assume that there is a feasible solution $[3 \ 1 \ 1 \ 2 \ 1 \ 3 \ 2 \ 2 \ 3]$. Then the representation given in Fig. 5 will correspond to the process sequence provided in Fig. 6.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 1 | 2 | 1 | 3 | 2 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|

Figure 5 Instance for the representation of the job-shop scheduling problems

| | | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 3,1,3,4 | 1,1,2,3 | 1,2,3,7 | 2,1,1,6 | 1,3,1,5 | 3,2,1,3 | 2,2,3,9 | 2,2,2,8 | 3,3,2,9 |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|

Figure 6 The process sequence corresponding to the representation (in Fig. 5)

As for a feasible solution that has been generated, it must be transformed into a feasible activity schedule before it is used to solve the scheduling problem in practice. Such a process is called decoding. A feasible solution can be decoded into several schedules. That is to say, an operation sequence will correspond to multiple scheduling solutions. Then the optimization of the decoding process also means that a better scheduling solution [12] can be obtained from the operation sequence.

Optimization methods: Judge if the current operation can be inserted into the existing free segment. If yes, it will significantly shorten the time spent on the feasible

scheduling to obtain the optimal or the quasi-optimal scheduling solution.

In addition to the research on the micro problem of task scheduling, in the study of macro problems such as supply chain design and integrated manufacturing, the BFO algorithm is introduced and some improvements have been made. In view of these NP problems, a series of researches have been carried out, and the integration model of supply chain facility transfer and production planning based on dynamic unit manufacturing has been put forward [13, 14]. In the field of artificial intelligence, bacterial foraging optimization algorithm has also been proved an efficient, stable and competitive bionic intelligent optimization method [15].

3.3 The Application of the BFO-Based Cell Searching Strategy

On the formation of individual bacteria, some articles have been studied, and a matrix based bacterial foraging optimization algorithm tracking constraint treatment (MPANT) is proposed to solve the cell formation problem while considering cell load variations and a number of exceptional elements [16]. This section focuses on the improvement of individual bacteria in search strategy.

In the foraging process of a cell, the operation (or the cell-based searching strategy) made by the cell to determine its searching direction according to the information collected by itself is carried out based on the specific steps as below:

1) Determine the cell I that will make a cell-based search in the bacterial swarm.

2) Randomly choose two positions on the representation sequence of the feasible solution that represents the cell. The interval between these two positions on the representation sequence is called stable range, where the various operation sequences (or the representation) will remain unchanged in current search process. In other words, the searching will always be made in the direction that has been chosen randomly.

3) The scheduling sequence identified by the cell is divided into three parts: Except the stable range in the middle, both intervals beside the stable range are separately called Displacement Interval A and Displacement Interval B. Sequences in both of the intervals will be reorganized randomly one by one. In other words, displacement operation will be performed separately in Interval A and Interval B. Such a displacement also means that the cell is taking another step towards the chosen direction. Due to the adoption of the operation-based representation strategy, the representation sequence obtained after the random displacement operation is still a feasible solution.

4) Fitness evaluation on the current position of the cell. If new position is more advantageous, it can be considered that the movement made this time is correct. Then take the cell on the new position to replace the original one, and this is equivalent to a displacement made by the original cell. Otherwise, the cell will return to the original position, and it means that the moving operation this time is cancelled. In the study of location evaluation and displacement of bacteria, some academic articles put forward a multilevel threshold method based on feature selection to improve the

applicability and practicability of the optimal threshold technology in the BFO algorithm [17].

5) A pause made by an individual cell. Now it faces two choices: One is the re-selection of a random direction and make a cell-based search. The other is to cease the searching immediately. In this case, it will skip to the next cell that is performing the searching operation in the scheduling system and the algorithm in this round ends now.

3.4 The Improved Searching Strategy Based on the Quorum Sensing Mechanism

The optimization of high-dimensional space is a difficult problem in optimization. With the increase of the problem dimension, the difficulty of finding the optimal solution increases exponentially. Although swarm intelligence algorithm has a strong advantage in optimization problems, the speed and accuracy of algorithm are still greatly reduced in the face of high-dimensional problems. For complex high-dimensional optimization problems, the fractal dimension optimization method can be used to fine divide the problems [18, 19]. The improved method makes full use of the change information of the local dimension, judges and updates the location information by dimension, and effectively improves the efficiency and accuracy of the algorithm.

Searching strategy can be improved based on the quorum sensing mechanism according to the specific steps as below:

1) Determine the cell I_{c_best} that occupies the currently best position in the bacterial swarm, and save the position and the fitness value.

2) Determine the cell I that will make a search in the bacterial swarm.

3) In the sequence of the best cell I_{c_best} , randomly choose a number J_{rand} (or the job number). For a new cell to be generated, make sure to guarantee that this job number will be started on the same position of the best cell in the process sequence.

4) Traverse the best cell I_{c_best} to determine the position P^j of the job number J_{rand} in the best cells.

5) Start to adjust the cell I . Traverse the cell I to find out the initial position (or the position of the J_{rand}^{th} workpiece undertaking the first process in the current sequence) of J_{rand} in the process sequence. Then displace it with the job number in the position P^j .

6) Make a cell-based evaluation on the current position of the cell.

7) Make a swarm-based evaluation on the current position of the cell. Compared with the cell I_{c_best} that occupies the currently best position among the bacterial swarm, if the new position of the cell I is more advantageous, it can be considered that the movement made this time has made the cell swim to the best position in the swarm. Then the best position will be updated to the position of this cell and the optimal fitness value of the swarm will be updated accordingly. Otherwise, no operation will be performed. In order to more accurately describe the nature of the relationship between the bacterial attraction and rejection, the hyperbolic tangent function can be introduced to improve the population performance. In addition, in order to retain the primary features of the

eliminated bacteria, the roulette technique can also be applied to the migration phase [20].

8) A pause made by an individual cell. Now it faces two choices: One is the re-selection of a random direction and make a cell-based search. The other is to cease the searching immediately. In this case, it will skip to the next cell that is performing the searching operation in the scheduling system and the algorithm in this round ends now.

4 ALGORITHM IMPLEMENTATION

To solve the task scheduling problem by using the optimization algorithm, first, we need to analyse the characteristics of aeronautical maintenance, design the coding strategy, and then design the processing flow according to the algorithm. Some studies have used bacterial foraging optimization algorithm to solve the aviation maintenance task scheduling problem, and verified the rationality and feasibility of the algorithm [21]. In this paper, based on these studies, the bacterial foraging optimization algorithm is improved, and then the algorithm processing flow is redesigned.

Fig. 7 shows the processing flowchart with the use of the improved bacteria foraging algorithm as the scheduling optimization algorithm library.

Use this algorithm library to verify the practical problems and the various process information is provided in Tab. 4.

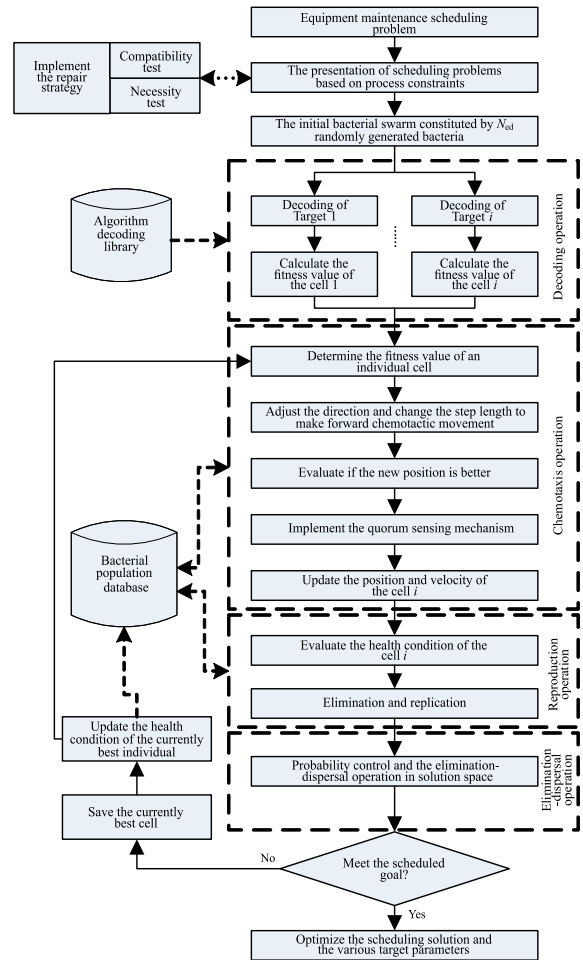


Figure 7 The processing flowchart for the optimization of scheduling algorithm library

Table 4 Some processes in the 400H scheduled maintenance of an aircraft

| To-be-repaired components | Process | Resources required | | | | | |
|------------------------------------|---|--------------------|----------------|----------------|----------------|----------------|----------------|
| | | M ₁ | M ₂ | M ₃ | M ₄ | M ₅ | M ₆ |
| J ₁ Altitude indicator | O ₁₁ Remove the altitude indicator | 2 | 3 | 4 | - | - | - |
| | O ₁₂ Power-on check | - | 3 | - | 2 | 4 | - |
| | O ₁₃ Install the altitude indicator | 1 | 4 | 5 | - | - | - |
| J ₂ Fire control system | O ₂₁ Remove the fire-control connector | 3 | - | 5 | - | 2 | - |
| | O ₂₂ Check the fuse protector | 4 | 3 | - | - | 6 | - |
| | O ₂₃ Fit the plug | - | - | 4 | - | 7 | 11 |
| J ₃ Steering system | O ₃₁ Remove flaps | 5 | 6 | - | - | - | - |
| | O ₃₂ Manipulation check | - | 4 | - | 3 | 5 | - |
| | O ₃₃ Install flaps | - | - | 13 | - | 9 | 12 |
| J ₄ Radar | O ₄₁ Remove the omni-directional warning unit | 9 | - | 7 | 9 | - | - |
| | O ₄₂ Infield test | - | 6 | - | 4 | - | 5 |
| | O ₄₃ Install the omni-directional warning unit | 1 | - | 3 | - | - | 3 |

Table 5 Process route

| To-be-repaired components | Process | Resources required | | | | | |
|------------------------------------|---|--------------------|----------------|----------------|----------------|----------------|----------------|
| | | M ₁ | M ₂ | M ₃ | M ₄ | M ₅ | M ₆ |
| J ₁ Altitude indicator | O ₁₁ Remove the altitude indicator | | 3 | | | | |
| | O ₁₂ Power-on check | | 3 | | | | |
| | O ₁₃ Install the altitude indicator | | 4 | | | | |
| J ₂ Fire control system | O ₂₁ Remove the fire-control connector | | | | | 2 | |
| | O ₂₂ Check the fuse protector | 4 | | | | | |
| | O ₂₃ Fit the plug | | | 4 | | | |
| J ₃ Steering system | O ₃₁ Remove flaps | 5 | | | | | |
| | O ₃₂ Manipulation check | | | | 3 | | |
| | O ₃₃ Install flaps | | | | | 9 | |
| J ₄ Radar | O ₄₁ Remove the omni-directional warning unit | | | 7 | | | |
| | O ₄₂ Infield test | | | | | | 5 |
| | O ₄₃ Install the omni-directional warning unit | | | | | | 3 |

Run the algorithm for 50 times and it converges at $f(T_j^*) = 17$ each time. After the search ends, an optimal process sequence $[O_{21}O_{11}O_{12}O_{13}O_{41}O_{31}O_{22}O_{23}O_{42}O_{32}O_{33}O_{43}]$ will be obtained with the process route provided in Tab. 5 and the relevant Gantt chart shown in Fig. 8.

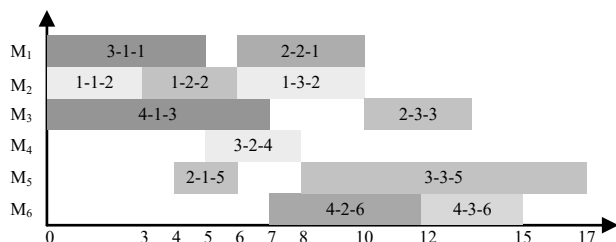


Figure 8 Gantt chart of scheduling results

5 CONCLUSIONS

Inspired by gene mutation, dynamically adjusts the tendency of bacteria to make the good bacteria more capable of action. By establishing bacterial quorum sensing mechanism, bacteria can be allowed to guide their swimming routes by using their peer experience, and enhance their global search capability. This improved bacterial foraging optimization algorithm can be applied to the optimization scheduling system for the aviation equipment scheduled maintenance. In this system, a complete airplane scheduled maintenance model can be established to efficiently solve the dynamic scheduling problem in the maintenance tasks. In addition, at the same time, it can realize the optimal combination and reasonable distribution of the various maintenance support resources.

6 REFERENCES

- [1] Mishra, S. (2005). A hybrid least square-fuzzy bacteria foraging strategy for harmonic estimation. *IEEE Trans on Evolutionary Computation*, 9(1), 61-73. <https://doi.org/10.1109/TEVC.2004.840144>
- [2] Passino, K. M. (2002). Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control System Magazine*, 22(3), 52-67. <https://doi.org/10.1109/MCS.2002.1004010>
- [3] Datta, T. & Misra, I. S. (2008). Improved Adaptive Bacteria Foraging Algorithm in Optimization of Antenna Array for Faster Convergence. *Progress in Electromagnetics Research C*, 1(1), 143-157. <https://doi.org/10.2528/PIERC08011705>
- [4] Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks / Perth, WA, Australia, 1942-1948*. <https://doi.org/10.1109/ICNN.1995.488968>
- [5] Eberhart, R. C. & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science / Nagoya, Japan*, 39-43. <https://doi.org/10.1109/MHS.1995.494215>
- [6] Shi, Y. H. & Eberhart R. C. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation / Alaska, USA*, 69-73. <https://doi.org/10.1109/ICEC.1998.699146>
- [7] Chu, Y., Mi, H., Ji, Z., & Wu, Q. (2010). Fast Bacterial Swarming Algorithm Based on Particle Swarm Optimization. *Data Acquisition & Processing*, 25(4), 442-448.
- [8] Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: Theoretical foundations, analysis, and applications. *Foundations of Computational Intelligence*, 3, 23-55. https://doi.org/10.1007/978-3-642-01085-9_2
- [9] Wang, S. F. & Zou, Y. R. (2003). Techniques for the Job Shop Scheduling Problem: A Survey. *Systems Engineering - Theory & Practice*, 1(1), 49-50.
- [10] Wang, W. Y., Tu, H. N., Xia, F. C., & Ma, Z. B. (2009). Bacterial Forging Algorithm for Job-shop Scheduling System. *Modern Manufacturing Technology and Equipment*, 189(2), 7-8.
- [11] Liang, Y. C. (2009). *Theory and Application of Swarm Intelligent Optimization Algorithm*. Science Press, Beijing, China.
- [12] Zhang, N. (2007). *Study on Job-Shop Scheduling Problems Based on Bacteria Forging Optimization Algorithm*. Changchuan: Jilin University.
- [13] Liu, C. F., Wang, J. F., & Leung, J. Y. T. (2018). Integrated bacteria foraging algorithm for cellular manufacturing in supply chain considering facility transfer and production planning. *Applied Soft Computing*, 62, 602-618. <https://doi.org/10.1016/j.asoc.2017.10.034>
- [14] Liu, C. F., Wang, J. F., Leung, J. Y. T., & Li, K. (2016). Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. *International Journal of Production Research*, 54(3), 923-944. <https://doi.org/10.1080/00207543.2015.1113328>
- [15] Liang, X. D., Lin, N., & Chen, H. N. (2016). Mobile robot dynamic path planning based on bacterial foraging behavior. *Chinese Journal of Scientific Instrument*, 37(6), 1316-1324.
- [16] Nouri, H. & Tang, S. H. (2013). Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations. *Journal of Manufacturing Systems*, 32(1), 20-31. <https://doi.org/10.1016/j.jmsy.2012.07.014>
- [17] Tang, K., Xiao, X., Wu, J., Yang, J., & Lou, L. (2017). An improved multilevel thresholding approach based modified bacterial foraging optimization. *Applied Intelligence*, 46(1), 214-226. <https://doi.org/10.1007/s10489-016-0832-9>
- [18] Jiang, J. G., Zhou, J. W., Zheng Y. C., & Wang, T. (2015). Adaptive bacterial foraging optimization algorithm. *Journal of Xidian University*, 42(1), 75-81.
- [19] Li, J. & Dang, J. W. (2017). Application of improved bacterial foraging algorithm in high-dimensional optimization problems. *Computer Science*, 44(4), 269-274.
- [20] Chen, Y. P., Li, Y., Wang, G., Zheng, Y. F., Xu, Q., Fan, J. H., & Cui, X. T. (2017). A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications*, 83, 1-17. <https://doi.org/10.1016/j.eswa.2017.04.019>
- [21] Han, Q. T., Cao, W. J., & Yang, L. (2012). Study on optimization of aeronautical maintenance process. *IEEE International Conference on Computer Science and Electronics Engineering / Hangzhou, China*, 203-206. <https://doi.org/10.1109/ICCSEE.2012.398>

Contact information:

Jia CUI
Naval Aeronautical University,
Dept. 3, No. 188, Ermalu Rd., Zhifu District,
Yantai, 264001, China
jiacui_cj@sina.com