International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Europeen des Systemes Automatises
Vol. 52, No. 1, February, 2019, pp. 49-55
Journal homepage: http://iieta.org/Journals/jesa

# Emergency Surgery Scheduling under Urban Emergencies Based on Improved Moth-flame Optimization

Feng Yang[1], Minghua Shi[2*]

[1] College of Management, Henan University of Chinese Medicine, Zhengzhou 450046, China
[2] College of Finance and Mathematics, West Anhui University, Lu'an 237012, China

Corresponding Author Email: minghuashi@163.com

https://doi.org/10.18280/jesa.520107

**Received:** 3 December 2018
**Accepted:** 19 January 2019

***Keywords:***
*emergencies, surgery scheduling, fatigue effect, moth-flame optimization (MFO), chaotic perturbation*

## ABSTRACT

This paper analyzes the features of emergency surgeries under urban emergencies, and establishes a model for batch surgery scheduling, taking the issue as the flow-shop scheduling problem (FSP) with parallel machines. The modelling also takes account of the fatigue of doctors after long-term continuous operations, which causes the aging of the operating time and the success rate of saving lives. Then, the improved moth-flame optimization (IMFO) was adopted to solve the established emergency surgery scheduling model. Finally, the proposed model and algorithm were verified through an actual case.

## 1. INTRODUCTION

In recent years, the continuous expansion of the city scale, coupled with the constant influx of migrants, has posed a serious threat to public security in urban areas [1]. Due to the high density of buildings and the narrow spaces in cities, it is highly difficult to implement rescue operations against emergencies. Under the co-scheduling of urban emergency rescue management system, the injury conditions must be identified and classified: those with light injuries should be treated at the site, while those with serious injuries should be transferred immediately to the hospital for surgery and first aid [2]. However, medical institutions are often overwhelmed by the sheer number of seriously injured people in urban emergencies. Against this backdrop, the urban public safety emergency management system should improve the efficiency of emergency surgery, trying to save the lives of the injured as much as possible through highly efficient utilization of limited medical resources.

The existing optimization plans for hospital operating room mainly focus on the optimization of surgical scheduling. For example, Luo et al. established a mixed integer programming model for the balanced use of the operating room, aiming to maximize the utilization of the operating room [3]. Considering the uncertainty of the operating time, Wang et al. [4] created a deterministic model of the surgical scheduling problem to maximize the hospital benefits, and proposed a two-stage robust optimization method for the interval operation scheduling problem. In light of the uncertainty of the surgical procedure, Denton et al. [5] established a robust optimization model to minimize the operating cost, and determined the upper and lower bounds of the optimal number of open operating rooms. Zhong et al. [6] described surgical scheduling as a parallel machine scheduling problem, and solved it by a two-stage approach.

These studies mainly consider how to reduce the operating time through maximal use of operating room resources under normal surgeries. However, there is no report on minimizing the operating time and maximizing the resource utilization facing the numerous seriously injured people under emergency surgeries.

This paper identifies the features of emergency surgery under urban emergencies, and views emergency surgery scheduling as a flow-shop scheduling problem (FSP) with parallel machines. On this basis, the batch surgery scheduling was modelled, considering the fatigue of doctors performing long-term continuous emergency surgeries. The fatigue problem causes the aging of the operating time and the success rate of saving lives. Then, the emergency surgery model was solved by the improved moth-flame optimization (IMFO) algorithm. Finally, the model and algorithm were proved effective through empirical analysis.

## 2. EMERGENCY SURGERY SCHEDULING PROBLEM

### 2.1 Problem description

The theories and methods of production scheduling have not been widely applied in the service industry, despite the extensive implementation in the manufacturing industry. In fact, the production scheduling theory in the medical service industry needs to be improved more urgently. Based on the production scheduling theory, Bai et al. [7] reviewed the relevant studies on production scheduling theory and operating room scheduling, analyzed the applicability and difference of production scheduling theories in the operating room, and proposed the operating room scheduling framework. Pham et al. [8] regarded the optimization problem as a multi-module shop scheduling problem, and established a corresponding mixed integer linear programming model to reduce the effects of performance indices on operating room timeout. Zhong et al. [9] managed operating room scheduling based on such theories as operational research, combinatorial optimization and scheduling theory.

In the daily hospital operation, there is basically no surgery on lots of injured people. However, once an emergency occurs, many injured people will be received to the hospital, waiting for surgery. Most of the injured have the same type of injuries, and thus requires similar surgeries. This type of emergency surgeries bears high resemblance with the processing of manufacturing jobs on machines. Therefore, the emergency surgery scheduling can be viewed as job-shop scheduling. Currently, many hospitals are ready to separate anesthesia preparation room from the anesthesia recovery room, forming three independent procedures. As shown in Figure 1, the three procedures are preoperative preparation (including anesthesia), surgery, and postoperative cleanup (including anesthesia recovery). When there is a need for surgery, the surgical scheduling center assigns the surgeries to each operating room by certain rules, and arranges the sequence between the surgeries. For each surgery, the total length covers three phases: preoperative preparation (including anesthesia), surgery, and postoperative cleanup (including anesthesia recovery). From the perspective of production scheduling, the entire scheduling process of emergency surgeries can be considered as the FSP. With more than one anesthesia preparation rooms, the operating rooms and anesthesia recovery rooms, the hospital is similar to a parallel machine job-shop. Each type of rooms is comparable to a type of machines that can be processed in parallel, while each injured person as a job. Thus, the emergency operation scheduling problem can be regarded as the flexible FSP, indicating that the job-shop scheduling theories can be introduced to emergency surgery scheduling.

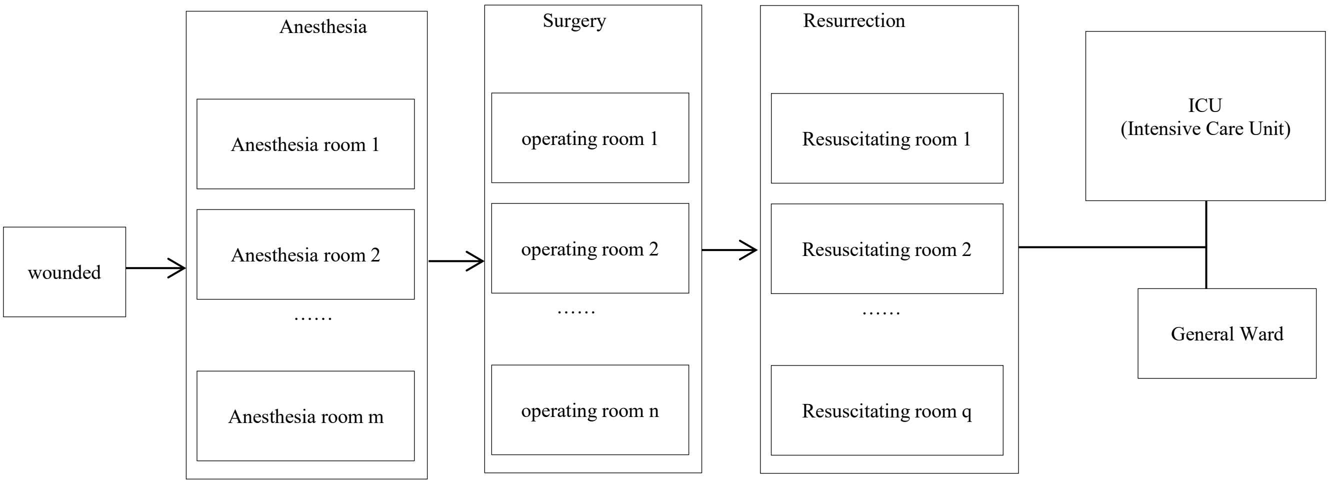

**Figure 1.** Emergency surgical procedures

### 2.2 The aging effect of fatigue problem

The urban emergency is usually associated with a huge number of injured people, pushing up the workload of surgeons. Under the high pressure, the doctors will inevitably suffer from fatigue, owing to such factors as gender, age and physical fitness. The fatigue problem has an aging effect on emergency surgery scheduling. Recent years has seen much attention being paid to scheduling problems with aging effect. In the earliest aging effect model, the processing time of each job is a non-decreasing linear function determined by the start time of the job.

In addition, many aging effect models have been developed based on job position [10]. Nonetheless, the existing aging effect models have a common drawback, that is, the job with a late start time and a backward processing position needs an infinitely long processing time, provided that the start time or processing position of the jobs are sufficiently large. This goes contrary to the actual production law. The same problem exists in emergency surgery scheduling [11-14]. Thus, a upper bound must be included in the research of aging effect, such that the actual surgery of an injured person will not last for an infinitely long time with the increase in the start time or queue position. Considering the above, this paper explores the time aging effect caused by doctor fatigue in emergency surgeries, and builds a realistic emergency surgery scheduling model with this effect as the constraint. The aging effect model can be expressed as:

$$C_{ijk}^{'} = C_{ijk} + \alpha_{ik} S_{ijk} \tag{1}$$

where $C'_{ijk}$ and $C_{ijk}$ are the actual treatment time and theoretical treatment time of the $j$-th surgical procedure of the $i$-th injured person on the $k$-th machine, respectively; $\alpha_{ik}$ is the aging coefficient of the $i$-th injured person on the $k$-th machine; $S_{ijk}$ is the start time for the $j$-th surgical procedure of the $i$-th injured person on the $k$-th machine [15].

### 2.3 Emergency surgery scheduling model

The emergency surgery dispatching problem can be described as follows: There are $n$ injured people $p_i$ ($i$=1, 2, …, $n$) to receive three surgical procedures, following the same sequence. The $j$-th surgical procedure contains $S_j$ parallel surgical beds ($S_j$>0, $j \in \{1,2,3\}$). For all injured people, the $j$-th surgical procedure can be implemented on any of the $S_j$ parallel surgical beds, and each injured person has a unique surgical time on the bed. Note that $C_{ijk}^{'}$ and $S_{ijk}$ are the actual treatment time and theoretical treatment time of the $j$-th surgical procedure of the $i$-th injured person on the $k$-th machine, respectively. The scheduling problem is subjected to the following hypotheses:

(1) All injured people share the same emergency surgical sequence, consisting of the same procedures.

(2) There is no priority limit for each injured person, i.e. no limit on which injured person should be subjected to surgery first.

(3) All injured people can receive surgery immediately after reaching the hospital.

(4) Each surgical procedure of each injured person can only be implemented on one surgical bed.

(5) Each surgical bed can only treat one injured person at the same time.

(6) The surgery cannot be interrupted once it begins.

(7) None of the surgical beds is faulty.

(8) The doctors will feel fatigued after operating for too much time, leading to operation time aging.

The optimization aims to minimize the maximum makespan:

$$\min C_{\max}=\min\{\max\{C_i^{'}\}\},\ i=1,2,...n \tag{2}$$

where, $C_i^{'}$ is the actual completion time of the surgery on the injured person $p_i$. There exists the fatigue-induced aging relationship between the actual completion time and the theoretical completion time $C_i$ ($i=1,2,...n$).

## 3. IMFO DESIGN

### 3.1 Introduction to the MFO

The MFO [16] is a heuristic optimization algorithm proposed by Mirjalili. In this algorithm, the moths in the matrix *M* are candidate solutions, and the corresponding fitness values are saved in the array *OM*. Let *n* be the scale of moth population and *d* be the dimension of the optimization problem. Then, the matrix and the array can be expressed as:

$$M=\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,d} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,d} \\ \cdots & \cdots & \cdots & \cdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,d} \end{bmatrix} \tag{3}$$

$$OM=\begin{bmatrix} OM_1 & OM_2 & \ldots & OM_n \end{bmatrix}^T \tag{4}$$

Flame is another core component in the algorithm. The matrix of flames is denoted as *F*, which has the same dimensions with matrix *M*. The corresponding fitness values are saved in the array *OF*. Then, the matrix and the array can be expressed as:

$$F=\begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & F_{1,d} \\ F_{2,1} & F_{2,2} & \cdots & F_{2,d} \\ \cdots & \cdots & \cdots & \cdots \\ F_{n,1} & F_{n,2} & \cdots & F_{n,d} \end{bmatrix} \tag{5}$$

$$OF=\begin{bmatrix} OF_1 & OF_2 & \ldots & OF_n \end{bmatrix}^T \tag{6}$$

Both moths and flames are candidate solutions. The only difference lies in the position update mechanism during evolution: each moth moves near the search space, driving the flames to approach the optimal position (candidate solution). In this way, the moth can find its optimal solution. The MFO is similar to a globally optimal triple of the optimization problem:

$$MFO=(I,P,T) \tag{7}$$

where, *I* is a function that produces a random moth population and its fitness values; *P* is a function that makes the moths move in the search space, such as to update the moth positions and return the updated matrix *M*; *T* is the judgement function of the termination condition. The three functions are expressed in equations (8)~(10), respectively.

$$I:\varphi\rightarrow\{M,OM\} \tag{8}$$

$$P:M\rightarrow M \tag{9}$$

$$T:M\rightarrow\{true,false\} \tag{10}$$

After initialization, function *I* iterates until returning *true*. To simulate the spiral flight features of moths, the relative position of each moth to each flame can be defined as:

$$M_i=S(M_i,F_j)=D_i\cdot e^{bt}\cdot\cos(2\pi t)+F_j \tag{11}$$

where, $M_i$ is the *i*-th moth; $F_j$ is the *j*-th flame; *S* is the spiral function; $D_i$ is the distance between the *i*-th moth and the *j*-th flame; *b* is a constant that defines the logarithmic spiral function; *t* is a random number in [-1, 1] that defines how close the moth approaches the flame at the next position (the closest distance and farthest distance are respectively depicted as t=-1 and t=1). The value of $D_i$ can be obtained by:

$$D_i=|F_j-M_i| \tag{12}$$

The author put forward an adaptive flame update mechanism to reduce the number of flames, thus speeding up the convergence of the algorithm:

$$Flame_N=round\left(N-l*\frac{N-1}{T}\right) \tag{13}$$

where, $Flame_N$ is the number of flames being reduced adaptively; *l* is the number of the current iteration; *N* is the maximum number of flames; *T* is the maximum number of iterations.

### 3.2 MFO based on chaotic perturbation

Like many other algorithm, the MFO often falls into the local optimum trap when it is adopted to solve real-world optimization problems. An effective way to improve the MFO is to improve the diversity of the population, such that the population maintains the ability of continuous optimization throughout the evolution process.

Chaos is a unique aperiodic motion in nonlinear systems. The chaotic behaviors are random and complex, but have certain inherent regularity. The basic idea of chaotic perturbation is as follows: mapping the optimization variables to the value interval of chaotic variable space by chaotic mapping rules; searching for the optimal solution using the ergodicity and randomness of chaotic variables; converting the acquired optimal solution into the optimization space. In this paper, the MFO is improved based on the chaotic sequence generated by the logic self-mapping function below:

$$L_{n+1,d}=1-2L_{n,d}^2,\quad n=01,2,...,\infty,\ L_{n,d}\in(-1,1) \tag{14}$$

Chaos will occur whenever the initial value of the iteration is not zero. The mapping domain is (-1, 1), excluding 0 and 0.5. *d* refers to the spatial dimension.

The MFO was improved based on chaotic perturbation through the following steps. Firstly, the *i*-th moth is located at position *x* in the *D*-dimensional space; each dimension in the spatial position of the moth needs to be mapped to [-1, 1] by equation (15) according to the nature of the logical self-mapping function. Secondly, the carrier operation is performed by equation (14); the resulting chaotic variables are introduced to the optimization variables, and used to search for the optimal solution; in this way, the new individual can be obtained through chaotic operation. Finally, the obtained chaotic variable sequence is transformed into the original solution space by equation (16); if a better solution is found, it should replace the original position of the *i*-th moth; otherwise, the chaotic search needs to be continued until reaching the pre-set number of searches.

$$L_{i,d}=2\cdot\frac{y_{id}-\alpha_{id}}{b_{id}-\alpha_{id}}-1,\quad d=1,2,\ldots,D \tag{15}$$

$$x_{id}^{'}=\frac{1}{2}(b_{id}-\alpha_{id})\cdot L_{id}+\frac{1}{2}(b_{id}+\alpha_{id}),\quad d=1,2,\ldots,D \tag{16}$$

where, $\alpha_{id}$ and $b_{id}$ are the upper and lower bounds of the *d*-dimensional variable of individual moth.

### 3.3 Coding design

The commonly used coding methods are not applicable to emergency surgery scheduling, for the problem aims to select a surgical bed from multiple parallel beds for a surgical procedure of each injured person. Thus, this paper proposes a matrix coding method to store feasible scheduling solution information, which saves moth positions as vectors in a matrix. Each feasible solution to the problem has two attributes: surgical bed selection from multiple parallel beds for a surgical procedure of each injured person, and the surgical sequence of each injured person. Suppose there are *n* injured people, denoted as $p_i$ ($i=1,2,...n$), to receive surgeries with *m* procedures ($S_1$, $S_2$, …, $S_m$). For emergency surgeries, it is obvious that, when m=3, there are $S_i$ parallel surgical beds for the i-th procedure. Then, a feasible scheduling plan of the problem can be described as:

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix},$$
$$XT=\begin{bmatrix} xt_{11} & xt_{12} & \cdots & xt_{1n} \\ xt_{21} & xt_{22} & \cdots & xt_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ xt_{m1} & xt_{m2} & \cdots & xt_{mn} \end{bmatrix} \tag{17}$$

where, *X* is the matrix of surgical beds, with $x_{ij}$ (1≤i≤m, 1≤i≤n) being the serial number of surgical beds for the *i*-th procedure of injured person $p_j$; *XT* is the matrix of surgical sequences, with $xt_{ij}$ being the sequence of surgical beds for the *i*-th procedure of injured person $p_j$. If $x_{ij}=x_{ik}$ ($j\neq k$), then injured person $p_j$ selects the same surgical beds for the *i*-th procedure.

The treatment time of the injured people on each surgical bed is also stored in a matrix. There are $S_i$ parallel surgical beds in each surgical procedure. For convenience, the surgical beds are numbered in turn as 1, 2, ..., $\lambda$ ($\lambda=\sum_{i=1}^{m}S_i$). For any surgical procedure *k* ($1\leq k\leq m$), the serial number $\omega$ of parallel surgical beds $M_\omega$ falls in the range below:

$$\omega=\begin{cases}[1,S_1], & k=1\\ [\sum_{i=1}^{k-1}S_i+1,\sum_{i=1}^{k}S_i], & x<k\leq m\end{cases} \tag{18}$$

For example, there were five wounded awaiting surgery. During the operation, there are 2 anesthesia preparation beds, 1 operation room and 3 anesthesia resuscitation beds. Then the number of operation beds in the three operation processes are {1,2}, {3}, {4,5,6} respectively. A feasible scheduling scheme is as follows:

$$X=\begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 5 & 6 & 4 & 5 \end{bmatrix},\quad XT==\begin{bmatrix} 2 & 1 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 & 5 \\ 1 & 2 & 1 & 2 & 1 \end{bmatrix}$$

From the matrix *X*, $x_{11}=x_{13}=x_{15}=1$ indicates that the first operating procedures of wounded persons $p_1$, $p_3$ and $p_5$ are anaesthetized on the operating bed. From the matrix *XT*, $xt_{11}=2$, $xt_{13}=1$, $xt_{15}=3$, which indicates that the queuing order of wounded $p_1$, $p_3$ and $p_5$ on the operating bed $M_1$ is 2→1→3. That is, $p_1$ is anesthetized first, then $p_3$, and finally $p_5$.

### 3.4 Population initialization

The initial population of the standard MFO is generated by random. In this paper, the moth information is initialized in the solution space, and the initial position information of some moths is optimized to improve the quality of the initial population. Each feasible solution contains two attributes, namely, the surgical bed selection from multiple parallel beds and the surgical sequence of the injured people. The information of the two attributes must be initialized at the same time. The initial information of surgical bed selection is randomly generated from the range of the serial number of parallel surgical beds available for each procedure, as specified in equation (18), aiming to ensure the feasibility of the randomly generated scheduling plan.

The surgical sequence of the injured people is initialized as vectors, whose values are all 1, and then modified according to the surgical bed selection information. The modification is a two-step process: converting all information on surgical bed selection into a scheduling matrix $X=[x_{ij}]_{m\times n}$; determining the surgical sequence values of the injured people according to the following method:

(1) For $x_{ij}=x_{ik}$ ($j\neq k$), if $i=1$, i.e. *i* is the first surgical procedure, then the surgical sequence of the injured people should be ranked in ascending order by the surgical time;

(2) Otherwise, the surgical sequence of the injured people should be determined by ranking the completion time of the previous procedure of each injured person in ascending order.

### 3.5 Implementation steps

The IMFO-based emergency surgery scheduling contains the following steps:

Step 1: Determine equation (2) as the fitness function of the algorithm.

Step 2: Initialize the algorithm parameters. Set the size of the moth population *n*, the maximum number of iterations *T*, the search space dimension *d*, the maximum number of flames *N*, and the current number of iterations *η=0*. Configure the coding of moths and flames according to the coding design, and initialize the population.

Step 3: Calculate the individual fitness of individual moth, find and save the current best position of individual moth. Determine whether the termination condition is satisfied. If yes, go to Step 9; otherwise, go to Step 4.

Step 4: Perform iterations. Update the number of flames by equation (13), compute the distance between the flame and the moth, and update the moth-flame position by equation (11).

Step 5: Calculate the fitness of individual moth, and save the positions of the moths and flames using equations (4) and (6), respectively.

Step 6: Find the current best position of individual moth. If it is better than the position reserved in the previous iteration, then save the current flame position as the optimal position.

Step 7: Evaluate the moth population, select the best 20% moths as the elite individuals, and perform chaos optimization by equations (14)~(16). Meanwhile, select the worst 20% of moths and replace them with randomly generated new individuals.

Step 8: Recalculate the individual fitness of each moth according to the position of the moth after moving. If the search accuracy or the maximum number of searches is reached, go to Step 9, otherwise go to step 3.

Step 9: Determine whether the termination condition is satisfied. If yes, go to Step 10; otherwise, let *η=η+1* and perform Steps 4~8.

Step 10: Output the final position of the flame and the corresponding fitness, forming the emergency surgery scheduling plan. Terminate the algorithm.

## 4. EXPERIMENTAL VERIFICATION

The proposed IMFO was verified by an emergency in a Chinese city. In this emergency, 12 patients with bone fractures are waiting for emergency surgery. All of them need to go through three surgical procedures: anesthesia, surgery and recovery. There are 2 beds in the anesthesia room, 5 tables in the operating room, and 3 beds in the recovery room. The anesthesia time of the patients differs by 1~2min because of the conditions of the doctor and patient; the surgical time of the patients differs within 10min; the recovery time of the patients differs within 10min (Conclusion the data of Shanghai Xinhua hospital were analyzed). The specific treatment time is listed in Table 1 below.

**Table 1.** The treatment time of the injured people in each surgical procedure (min)

| Patient | $S_1$ | | $S_2$ | | | | | $S_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | $M_8$ | $M_9$ | $M_{10}$ |
| $p_1$ | 2 | 2 | 60 | 60 | 58 | 58 | 57 | 27 | 28 | 27 |
| $P_2$ | 4 | 2 | 58 | 56 | 59 | 59 | 58 | 30 | 29 | 30 |
| $P_3$ | 3 | 2 | 58 | 57 | 61 | 58 | 59 | 31 | 34 | 29 |
| $P_4$ | 2 | 2 | 59 | 61 | 58 | 60 | 58 | 33 | 35 | 32 |
| $P_5$ | 2 | 3 | 58 | 57 | 58 | 57 | 60 | 32 | 33 | 28 |
| $P_6$ | 3 | 2 | 57 | 63 | 65 | 58 | 61 | 35 | 36 | 35 |
| $P_7$ | 3 | 2 | 56 | 57 | 59 | 59 | 62 | 29 | 30 | 29 |
| $P_8$ | 3 | 2 | 57 | 66 | 66 | 58 | 58 | 32 | 35 | 34 |
| $P_9$ | 2 | 4 | 58 | 59 | 62 | 60 | 59 | 33 | 32 | 31 |
| $p_{10}$ | 4 | 2 | 59 | 63 | 65 | 62 | 58 | 35 | 34 | 33 |
| $p_{11}$ | 2 | 3 | 58 | 57 | 58 | 59 | 60 | 29 | 28 | 28 |
| $p_{12}$ | 3 | 3 | 59 | 64 | 62 | 60 | 61 | 30 | 29 | 32 |

The emergency surgery scheduling aims to minimize the maximum surgical completion time of the injured people. For comparison, the IMFO was contrasted with the classic MFO, the particle swarm optimization (PSO) and the cuckoo search (CS) algorithm. The numbers of moths, flames, particles and nests were all set to 30, and the maximum number of iterations was set to 500. The parameter settings of each algorithm are given in Table 2. The simulation was carried out on Windows 7, Intel® Core™ i3-2350M Processor (2.0GHz), 2GB memory and Matlab 2015b.

**Table 2.** Parameter settings of each algorithm

| Algorithm | Parameters |
|---|---|
| IMFO | $b = 1, t \in [-1,1]$ |
| MFO | $b = 1, t \in [-1,1]$ |
| PSO | $w = 0.9, c_1 = c_2 = 2$ |
| CS | $P_a = 0.25$ |

The mean value was obtained after 20 random simulations. The results are recorded in Table 3 below.

**Table 3.** Maximum completion time and mean value of each algorithm after 20 simulations

| Algorithm | | IMFO | MFO | PSO | CS |
|---|---|---|---|---|---|
| Target value after 20 simulations | 1 | 210 | 240 | 258 | 210 |
| | 2 | 210 | 210 | 240 | 230 |
| | 3 | 210 | 234 | 246 | 210 |
| | 4 | 210 | 210 | 210 | 236 |
| | 5 | 210 | 214 | 210 | 236 |
| | 6 | 210 | 210 | 220 | 210 |
| | 7 | 210 | 216 | 224 | 210 |
| | 8 | 210 | 218 | 218 | 210 |
| | 9 | 210 | 217 | 210 | 210 |
| | 10 | 218 | 210 | 252 | 226 |
| | 11 | 210 | 220 | 230 | 210 |
| | 12 | 220 | 210 | 210 | 230 |
| | 13 | 210 | 214 | 231 | 224 |
| | 14 | 210 | 210 | 242 | 224 |
| | 15 | 210 | 210 | 210 | 210 |
| | 16 | 216 | 210 | 216 | 226 |
| | 17 | 210 | 210 | 224 | 225 |
| | 18 | 219 | 230 | 231 | 223 |
| | 19 | 210 | 234 | 230 | 220 |
| | 20 | 210 | 234 | 245 | 219 |
| Mean value | | 211.65 | 218.45 | 227.85 | 220.15 |

As shown in Table 3, the minimum value of the objective function converges to the extreme value as the population evolves. After 20 simulations, the IMFO reached the minimum 16 times, while the classic MFO only reached the minimum 9 times, indicating that the MFO with chaotic perturbation can effectively avoid the local optimum trap. In addition, the PSO reached the minimum 5 times and the CS reached the minimum 8 times. This means the MFO enjoys better global convergence than the PSO and the CS. To present a clear picture of the convergence speed of each algorithm, the convergence curve of each algorithm in the 13th simulation was plotted (Figure 2).

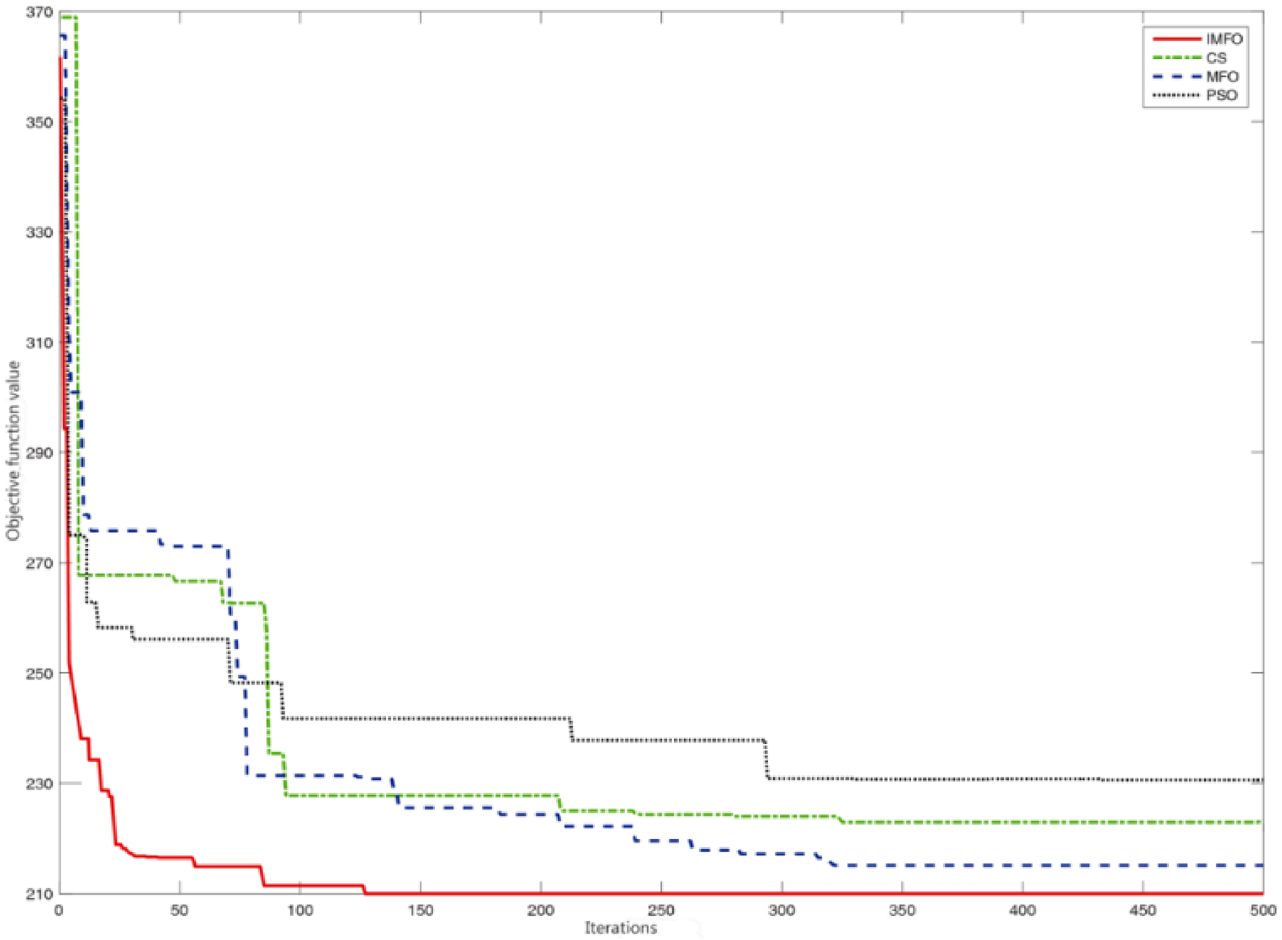

**Figure 2.** The optimal target value convergence curves of the four algorithms in the 13th simulation

It can be seen from Figure 2 that the IMFO converged to the optimal target value faster than the classic MFO, the PSO and the CS. Thus, the optimal scheduling plan can be obtained as:

$$X=\begin{bmatrix} 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 1 & 2 \\ 5 & 4 & 4 & 6 & 4 & 3 & 3 & 3 & 5 & 7 & 7 & 6 \\ 9 & 10 & 10 & 10 & 9 & 8 & 8 & 8 & 10 & 9 & 9 & 8 \end{bmatrix},$$

$$XT==\begin{bmatrix} 4 & 3 & 6 & 5 & 1 & 4 & 6 & 1 & 2 & 5 & 3 & 2 \\ 2 & 2 & 3 & 2 & 1 & 2 & 3 & 1 & 1 & 2 & 1 & 1 \\ 3 & 2 & 4 & 3 & 1 & 3 & 4 & 1 & 1 & 4 & 2 & 12 \end{bmatrix}$$

The Gantt chart of the parallel surgical bed selection for the problem is presented in Figure 3.

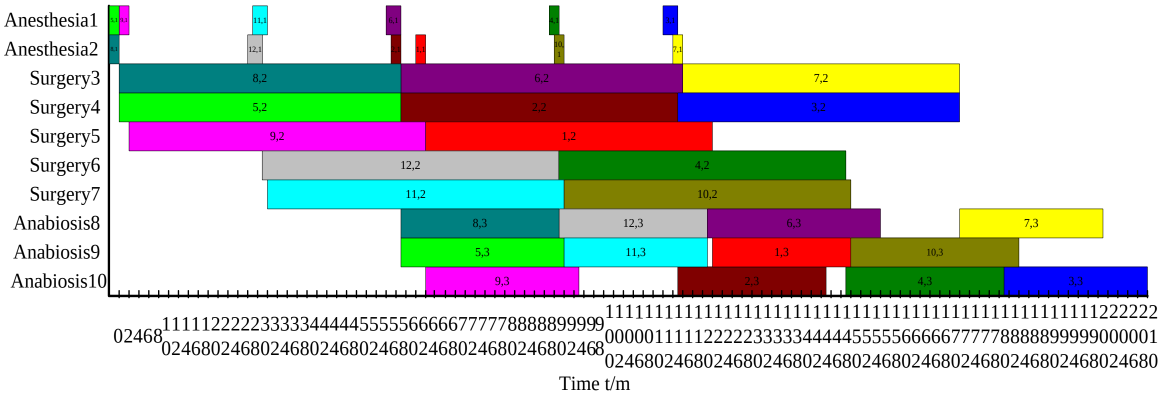

**Figure 3.** The optimal Gantt chart for emergency surgery scheduling

Figure 3 shows that the emergency surgery tasks of the injured people are relatively compact, and the treatment time is basically continuous. In this way, the total surgery time can be minimized, indicating that the scheduling results are reasonable.

## 5. CONCLUSIONS

In this paper, a surgery scheduling model is established in light of flexible FSP, considering the aging effect of fatigue among doctors after prolonged working. Then, the model was solved by the MFO with chaotic perturbation, the information of feasible scheduling solution was saved in matrix coding, the population was initialized, and the algorithm steps were specified. Finally, the proposed model and algorithm were verified through a case study on emergency surgeries for 12 patients with bone fractures.

The results show that the proposed emergency surgery scheduling model can describe emergency surgeries well, while the IMFO can output a rational scheduling plan and outperform the classic MFO, PSO and CS. Of course, the proposed solution only applies to single-objective optimization problems. The future research will explore the emergency surgery scheduling with multiple objectives and resource constraints.

## ACKNOWLEDGMENT

This work was supported by the Ministry of Education Humanities and Social Sciences Research Youth Fund Project(18YJCZH216) and the Research on Bidding for Henan Government Decision-making(2018B461) and the General Topics of the 13th Five-Year Plan for Educational Science in Henan Province([2018]-JKGHYB-0129) and the Natural Science Foundation of Anhui Province (Grant NO. 1908085QG306).

## REFERENCES

[1] Shu W, Luo L. (2008). Surgical scheduling research based on objective programming. Technology and Market (2): 42-44. http://dx.chinadoi.cn/10.3969/j.issn.1006-8554.2008.02.046

[2] Wang J, Liu HT, Huang J. (2017). Research on the route optimization of ambulance treatment and transportation after disaster based on the injured classification. Chinese journal of management science 25(8): 114-122. http://dx.chinadoi.cn/10.16381/j.cnki.issn1003-207x.2017.08.012

[3] Luo Y, Luo L, Zhou Y, You Y. (2016). Surgery scheduling comparative study based on MIP. Industrial Engineering and Management 21(2): 146-149. http://dx.chinadoi.cn/10.19495/j.cnki.1007-5429.2016.02.020

[4] Wang Y, Tang J. (2016). A two-stage robust optimization method for solving surgery scheduling problem. Journal of Systems Engineering 31(4): 431-440. http://dx.chinadoi.cn/10.13383/j.cnki.jse.2016.04.001

[5] Denton BT, Miller AJ, Balasubramanian HJ, Huschka T. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. Operations Research 58(4): 802-816. https://doi.org/10.1287/opre.1090.0791

[6] Zhong L, Luo S, Wu L, Xu L, Yang JH, Tang GC. (2014). A two-stage approach for surgery scheduling. Journal of Combinatorial Optimization 27(3): 545-556. https://doi.org/10.1007/s10878-012-9535-2

[7] Bai X, Luo L, Yang C. (2015). Application research framework of production scheduling theory in hospital operating room scheduling optimization. West China Medical Journal (10): 1990-1995. https://doi.org/10.7507/1002-0179.20150568

[8] Pham DN, Klinkert A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. European Journal of Operational Research 185(3): 1011-1025. https://doi.org/10.1016/j.ejor.2006.03.059

[9] Zhong LW, Luo SC, Yang GP, Mu ZL, Wang GY, Xu L, Zhao CM, Zhang P. (2009). Study on surgery scheduling optimization and computer program management in operating room. Journal of Shanghai Second Polytechnic University 26(4): 286-290. http://dx.chinadoi.cn/10.19570/j.cnki.jsspu.2009.04.005

[10] Liu CL, Wang JJ. (2017). Single-machine scheduling with deteriorating jobs, aging effect and rejection. Operations Research & Management Science 26(6): 95-101. https://doi.org/10.12005/orms.2017.0142

[11] Li H, Jiang D. (2013). Research on operating room scheduling problem based on partheno-genetic tabu search algorithm. Application Research of Computers 30(3): 699-702. http://dx.chinadoi.cn/10.3969/j.issn.1001-3695.2013.03.014

[12] Zhu Y, Zhang YL, Song M. (2015). Surgical scheduling considering setup time between surgeries and setup time between surgical teams. Journal of Southeast university (Natural Science Edition) 45(6): 1218-1222. http://dx.chinadoi.cn/10.3969/j.issn.1001-0505.2015.06.034

[13] Wei X, Jiao Y, Lim G. (2013). Modified ant colony algorithm for surgery scheduling under multiresource constraints. Advances in Information Sciences & Service Sciences 5(9): 810-818.

[14] Weiss G. (1991). Scheduling: Theory, algorithms, and systems. by Michael Pinedo. Interfaces 25(4): 130-132.

[15] Batun S, Begen MA. (2013). Optimization in healthcare delivery modeling: Methods and applications. Handbook of Healthcare Operations Management, Springer New York 75-119. https://doi.org/10.1007/978-1-4614-5885-2_4

[16] Mirjalili S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems 89: 228-249. https://doi.org/10.1016/j.knosys.2015.07.006