Cluster Computing (2022) 25:3591–3616
https://doi.org/10.1007/s10586-022-03589-0

An improved multi-objective imperialist competitive algorithm for surgical case scheduling problem with switching and preparation times

Hui Yu[1] · Jun-qing Li[2] · Xiao-long Chen[1] · Wei Niu[1] · Hong-yan Sang[2]

Received: 29 August 2021 / Revised: 13 March 2022 / Accepted: 18 March 2022 / Published online: 7 April 2022

Abstract
Surgical case scheduling is a key issue in the field of medician, which is a challenging work because of the difficulty in assigning resources to patients. This study regards the surgical case scheduling problem as a flexible job shop scheduling problem (FJSP). Considering the switching and preparation time of patients in different stage, an improved multi-objective imperialist competitive algorithm (IMOICA), which adopts the non-dominant sorting method, is proposed to optimize the whole scheduling. First, the social hierarchy strategy is developed to initialize the empire. Then, to enhance the global search ability of the algorithm, the concept of attraction and repulsion (AR) is introduced into the assimilation strategy. Moreover, to increase the diversity of the population, the revolution strategy is utilized. Finally, the variable neighborhood search (VNS) strategy is embedded to improve its exploitation capacity further. Experiments show that scheduling in advance saves time and cost, and IMOICA can solve the surgical case scheduling problem studied efficiently.

Keywords Surgical case scheduling · Imperialist competitive algorithm · Variable neighborhood search strategy

1 Introduction

Scheduling has been applied in many practical problems. Such as, in medical aspect, Burdett and Kozan [1] formulated medical scheduling to utilize various treatment spaces effectively. In environment aspect, Zhang et al. [2] considered optimizing scheduling to achieve the purpose of saving resources and protecting the environment. About cloud computing aspect, Chen et al. [3] considered that efficient scheduling approaches show promising ways to reduce the energy consumption of cloud computing platforms while guaranteeing quality of service requirements of tasks. In iron and steel production aspect, Tang et al. [4] introduced scheduling into the whole process of steel production to improve productivity and save energy.

✉ Hui Yu
1692816275@qq.com

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250000, People's Republic of China

[2] School of Information, Liaocheng University, Liaocheng 252059, People's Republic of China

Since the outbreak of COVID-19, the medical problem has become the main concern of people. Therefore, this paper takes the medical scheduling problem as the research focus. Medical scheduling includes physical examination scheduling [5], outpatient scheduling [6], nurse scheduling [7], surgical case scheduling [8] and so on.

Surgery is an important activity in most hospitals as it is estimated to generate about 2/3 of hospital revenue and consume 40% of hospital resource costs [9, 10]. Surgical case scheduling is a challenging problem faced by hospital managers. Consequently, many researchers have been motivated to study surgical case scheduling problem to save costs and improve resource utilization [11]. Especially in recent years, people are facing more and more health problems. The study of surgical case scheduling has become a research hotspot naturally.

In this paper, the surgical case scheduling problem is studied, taking into account all stages of hospitalization: pre-operative, peri-operative, and post-operative (Fig. 1). The patient flow during surgery can be described as follows:

- Pre-operative: When the patient enters the pre-operative holding unit (PHU), doctors and nurses are required to examine the patient's physical condition and do some preoperative preparation. Then, the patient is taken to the operating room (OR).
- Peri-operative: Upon arrival in the OR, the patient is anesthetized by the anesthesiologist. Then, surgery is performed by one or more surgeons with the assistance of one or more nurses.
- Post-operative: After surgery, general patients will be taken to the post-anesthesia care unit (PACU) for anesthesia recovery; mild patients may bypass PACU; and severe patients will be admitted to the intensive care unit (ICU) where professional doctors and nurses will take care of them, then went into a second surgery.

Compared with the traditional scheduling optimization problem, surgical case scheduling has its unique difficulties. It requires a variety of resources, and the availability of the required resources must be taken into account. At present, many general hospitals adopt the open scheduling strategy, which requires each OR to meet the needs of multi-departments. Open scheduling strategy increases the workload of the OR and further complicates the scheduling problem. If the schedule is too loose, the OR will be vacated, which will not only cause a waste of resources, but also make the number of operating tables unable to meet the actual needs of patients, resulting in patient dissatisfaction. If the schedule is too tight, it will lead to insufficient opening hours of surgery, overtime work for medical staff and other situations that will lead to high labor costs. Considering the availability of resources during surgery, the surgical case scheduling problem is regarded as the flexible job shop scheduling problem (FJSP). The patient is the job, the related resource is the machine, and the surgery stage is the operation.

With the deepening of medical reform and the limitation of income, the traditional mode of thinking which relies on "generating income" to help the development of hospitals has been impacted severely. The mode of using cost controllability to promote "saving expenditure" and promoting the development of medical units has been more and more recognized by everyone [12]. In recent years, cost has become one of the three major sectors of hospital competition. Many medical institutions have been developing tools to reduce costs and ensure the efficient use of hospital facilities [13–15].

Under the same results, early surgery, shorter surgery time, less exposure time, and less trauma time are beneficial for patient recovery and wound repair, including eventual swelling. Therefore, surgery time and cost are two important indexes concerned by hospitals [16, 17]. In the whole process of the surgery, the patient expects the surgery to be completed in the shortest time and the hospital wants to spend the lowest medical cost. This study takes the completion time of the surgery and medical cost of the whole surgery as the research object.

Keeping the above consideration in mind, this study proposes an improved multi-objective imperialist competitive algorithm, named IMOICA, to solve the surgical case scheduling problem. The main contributions can be described as the following: (1) the switching time and the preparation time constraints are introduced. Patients have different consumption in different stages, for example, as shown in Fig. 1, there is a time interval between the previous stage and the next stage, which can be divided into switching and preparation times. Switching time is the time spent by the patient at different stages. Preparation time can be seen as the patient's waiting time before formally proceeding to the next stage of treatment. In previous studies, these two processes were ignored, although ignoring these two processes would simplify the study, there would be a big difference between this and the actual scheduling. Therefore, Considering switching and preparation times here will make the scheduling developed in advance more realistic; (2) to generate a balanced initial population, a social hierarchy strategy is adopted in IMOICA, which can enhance the convergence speed of algorithm; (3) to enhance the global search capability of the algorithm, the concept of attraction and repulsion (AR) is introduced into the assimilation strategy; (4) to enhance the diversity of the population, the revolution strategy is introduced; (5) a variable neighborhood search (VNS) strategy is embedded in the proposed algorithm to improve the exploitation capability further.

The whole surgery process is studied and divided into different stages, which makes the research problem in line with the actual situation and simplify the whole research. The proposed algorithm is applied to the research problem, the scheduling sequence is optimized in the process of optimizing the algorithm.

The rest of this paper is organized as follows: Sect. 3 introduces the mixed integer linear programming (MILP) model. Section 4 describes the proposed algorithm with all of the components. The computational results and comparisons are reported in Sect. 5. Finally, Sect. 6 summarizes the conclusions of this study and discusses future research directions.

2 Related work

One of the most difficult problems in scheduling problems is the job shop scheduling problem (JSP) [18, 19]. A classic JSP similar to the real production situation is called FJSP. The FJSP consists of two sub-problems, i.e., routing and scheduling. The routing sub-problem is to assign each operation to a machine among a set of given machines, whereas the scheduling sub-problem is to sequence the assigned operations on all machines to obtain a feasible schedule with a satisfactory objective value. The general JSP is strongly NP-hard, while the FJSP is a much more complex version of the JSP, so the FJSP is strongly NP hard [20–25].

Various meta-heuristic algorithms have been proposed for the FJSP. Boukef et al. [26] studied a particle swarm optimization (PSO) algorithm to solve the FJSP. The PSO algorithm can converge quickly, solve problems efficiently, and provide accurate mathematical models. In the PSO algorithm, initial parameters may be difficult to define and may fall into local optimal. De Giovanni et al. [27] studied the genetic algorithm (GA) for the distributed FJSP. The GA has many functions and can solve complex problems, but it can be very time-consuming. The solution may be inaccurate and may fall into local optimal. Bagheri et al. [28] studied an artificial immune algorithm (AIA) for the FJSP. The AIA uses several strategies for generating the initial population and selecting the individuals for reproduction. Different mutation operators are also utilized for reproducing new individuals. The AIA can generate the optimal solution in short time, but the stability of the algorithm is greatly affected by the concentration of antibodies. Wang et al. [29] proposed an effective artificial bee colony (ABC) algorithm for solving the FJSP with the criterion to minimize the Makespan. The ABC algorithm stresses the balance between global exploration and local exploitation. Inspired by the intelligent foraging behavior of bees, the ABC can look for the best local and global solutions. However, it has the disadvantage of premature convergence of continuous search. For the FJSP with setup and transportation times, Rossi [30] proposed an ant colony optimization (ACO) algorithm to enhance the machine assigning/sequencing constraints and dynamic visibility function. The ACO algorithm derived from the performance of actual ants is compatible and can solve multiple problems at the same time, but the probability distribution of each iteration may change. Jiang and Zhang [31] proposed the grey wolf optimization (GWO) algorithm for the FJSP with typical discrete characteristics. The GWO algorithm has the characteristics of strong convergence performance, but it is easy to fall into local optimization and poor stability.

Atashpaz-Gargari and Lucas [32] proposed an optimization algorithm influenced by imperialist competitive, called the Imperial competitive algorithm (ICA). This was the first time the algorithm was proposed. The "mining" and "exploration" capabilities were not well balanced; therefore, researchers continue to optimize ICA in the follow-up. Talatahari et al. [33] presented a new chaos ICA. Although this algorithm improved the global optimization ability of the algorithm, but the search performance of the algorithm was reduced greatly. To solve the FJSP with transmission time, Karimi et al. [34] studied a hybrid ICA. In the algorithm, the local search strategy was used to enhance the local search ability of the algorithm, but the global search performance of the algorithm was not improved. Jian et al. [35] embed ant colony algorithm in ICA. In the process of assimilation, pheromone guidance mechanism was added to update the position of ants, which prevented the ICA solution from falling into local optimization. However, the complexity of the algorithm was increased greatly. Zhang et al. [36] proposed an improved ICA to solve the rebalancing problem of multi-objective bilateral assembly lines with space and resource constraints, without considering the conflict problem of ICA in solving multi-objective problems. For the multi-criteria engineering design, Mohamed et al. [37] designed a multi-objective ICA, although considering the problems of the algorithm in dealing with multi-objective problems, the local search ability of the algorithm was not been well improved. Therefore, in view of the shortcomings of the existing ICA, the improved multi-objective imperialist competitive algorithm, named IMOICA, is used to optimize the whole scheduling process.

In addition to FJSP, Meta-heuristic algorithm has also made great progress in big data text clustering. Mahdavi et al. [38] proposed a novel hybrid harmony search (HS) based algorithms for clustering the web documents that finds a globally optimal partition of them into a specified number of clusters. Mahdavi et al. [39] introduced a novel Harmony K-means Algorithm (HKA) based on HS optimization method for document clustering. García et al. [40] adopted a hybrid technology based on genetic population optimization algorithm and Nelder-Mead simplex search to solve the time-domain constrained data clustering problem. Jahwar and Abdulazeez [41] pointed out that k-means clustering algorithm can be applied to meta-heuristic algorithm to improve algorithm performance. Abualigah et al. [42] studied the application of optimization algorithm in text clustering, and studied the accessibility and application of appropriate optimization algorithms for each class. Irfan et al. [43] pointed out that to overcome the inherent limitations of learning classifier system-based systems to high-dimensional problems, the hybrid model designed by the integration of learning classifier system and deep learning methods has been a research hotspot in recent years. Irfan et al. [44] studied the application of brain-inspired lifelong learning model based on neural learning classifier system in underwater data classification and developed a continuous learning system like human beings to improve classification performance.

Unlike the FJSP with a single objective, the FJSP with multiple objectives has attracted researchers' attention recently. However, most of the literature on the multiple-objective FJSP adopts the aggregation approach. The drawback of that approach is that it generates only a single solution at each run. Thus, little information can be provided to the decision maker regarding the quality of each performance criterion. The Pareto-based approach has become a practical tool for solving the multi-objective FJSP recently. Lei proposed a Pareto archive particle swarm optimization [45], Li et al. presented a hybrid Pareto-based local search algorithm [46], Wang et al. conducted a Pareto-based estimation of distribution algorithm [47], and Gao et al. investigated a Pareto-based grouping discrete harmony search algorithm [48]. Furthermore, Wu and Sun proposed a non-dominated sorted genetic algorithm to solve the multiple-objective FJSP with two energy-saving measures [49]. Luo et al. developed an elaborately-designed multi-objective Grey wolf optimization algorithm and used two Pareto-based mechanisms to determine the leading wolf and the lowest (worst) wolf [50]. Li et al. [51] proposed an elitist non-dominated sorting hybrid algorithm for the multi-objective FJSP. Wang [52] used a hybrid multi-objective evolutionary algorithm based on decomposition to solve the multi-objective FJSP under time-of-use electricity price conditions, where all the optimal solutions of all single-objective sub-problems constituted the final Pareto set.

Sentiment analysis plays an important role in healthcare. Wallace et al. [53] generated a probabilistic model to analyze sentiments to capture patients' views on health care. To explore the role of social media in shaping the understanding of digital health care, Afyouni et al. [54] conducted sentiment analysis and found that people's overall view of digital health care is generally positive. Du et al. [55] studied the challenges and opportunities of sentiment analysis in medical settings and pointed out that sentiment analysis in medical literature requires a domain specific source of emotion and complementary context-dependent characteristics to correctly interpret implied emotions. Park and Woo [56] believed that sentiment analysis is the most common text categorization tool, which could be used to learn about gender, especially for people with sensitive diseases. Jiménez-Zafra et al. [57] used supervised learning and dictionary-based sentiment analysis to analyze online comments about drugs and doctors.

The quality of surgical case scheduling has a great impact on the sentiment of patients. FJSP is a strongly NP hard, the surgical case scheduling problem is also strongly NP-hard. Therefore, researchers have proposed different methods to optimize various problems related to surgical case scheduling. Pham and Klinkert [8] studied a new surgical case scheduling approach which uses a novel extension of the JSP called multi-mode blocking job shop (MMBJS) and discussed the use of the MMBJS model for scheduling elective and add-on cases. Cardoen et al. [58] used the MILP model to optimize the multi-objective surgical case scheduling problem to facilitate the decision-making process of the OR scheduler. The starting time of the surgery was determined by explicitly fixing the value of the variable by solving multiple knapsack problems. Considering the uncertainty of patients' hospitalization time and the availability of intensive care unit resources, Min and Yah [10] proposed a stochastic programming model, which assigns patients to the community to minimize the waiting time and overtime of the patient. Aiming at the multi-day, multi-resource, patient-priority-based surgery case scheduling problem, Vijayakumar et al. [59] presented a MILP model based on efficient First Fit Decreasing-based heuristic to increase the utilization rate of the OR. Lee and Yih [60] proposed a flexible job shop model with fuzzy sets, which takes into account patient waiting, idle clinical resources and total completion time in the process to reduce the delay in the flow of people in the OR. Cappanera et al. [61] developed a mixed-integer programming model to compare three different scheduling policies in the master surgical scheduling context with respect to three performance standards. Al Hasan et al. [62] presented a MILP model based on dictionary method to minimize the overtime of the surgical unit staff, the number of ORs used and the number of instruments processed in emergency in the sterilizing unit while respecting the current level of service represented by the total number of patients operated per month at the orthopedic surgery unit. Behmanesh and Zandieh [63] developed a novel bi-objective ant system to minimize Makespan and the number of unscheduled surgical cases simultaneously. Furthermore, these studies are analyzed, as shown in the following Table 1.

In recent years, researchers have optimized the problems related to different surgical case scheduling. Through the investigation, it is found that, first, most of the related studies on surgical case scheduling only involve the peri-operative stage of surgery. Only consider the process of the patient from entering the OR to the end of the surgery and ignored many details of the surgery process. This paper combines with the actual situation and studies the whole surgical process. The whole surgical process is divided into three different stages: pre-operative, peri-operative, post-operative, and the resource selection of different stages is pondered. Next, most of the previous studies are aimed at minimizing surgery time, and the medical cost is considered in our study on the basis of previous studies. In addition, according to the survey, it is found that the two important constraints of switching and preparation times are not studied in the previous surgical case scheduling

Table 1 Simple analysis of the surgical cases scheduling problem

	Contribution	Shortcoming
Pham and Klinkert [8]	A new multi-mode blocking job shop model was adopted to optimize the surgical case scheduling	The MMBJS model was not flexible enough to deal with additional cases of scheduling
Cardoen et al. [58]	They studied a MILP model to facilitate the decision process of the OR scheduler	The optimization process was not considered comprehensively, so it was difficult to implement it in reality
Min and Yih [10]	A stochastic programming model was proposed to minimize patient waiting time and overtime	The result was random, which was quite different from the actual scheduling
Vijayakumar et al. [59]	A MILP model based on efficient First Fit Decreasing-based heuristic was proposed to increase the utilization rate of the OR	The scheduling process studied was too general, which brings difficulties to the implementation of scheduling
Lee and Yih [60]	A flexible job shop model with fuzzy set was studied to reduce the delay in the flow of people in the OR	The model was solved by two-stage decision-making process, and errors was easy to occur in different stages of transformation
Cappanera et al. [61]	They presented a mixed-integer programming model to the master surgical scheduling for maximizing the number of surgeries scheduled and balance the beds and OR daily workloads	The hospital dimension was not explained, which would affect the efficiency of scheduling. In addition, some hospital resources (anesthesiologists, ICU beds, medical equipment) that have an important impact on the schedule were ignored
Al Hasan et al. [62]	A MILP model based on dictionary method was presented to minimize the overtime of the surgical unit staff and the number of instruments	Uncertainty in the actual duration of surgery due to multiple factors (e.g., surgical complications) and significant gaps between planned and realized schedules
Behmanesh and Zandieh [63]	For solving the surgical case scheduling with minimize makespan and the number of unscheduled patients, a MILP model is proposed	Ignoring the details of the stages would lead to a gap between the scheduling in advance and the actual scheduling

problem, and these two constraints are taken into account in this study. Finally, a MILP based on sequence is established, and the IMOICA is used to optimize the whole scheduling process.

3 Problem description

In the surgical case scheduling system includes three stages: pre-operative, peri-operative, and post-operative (Fig. 1). The patient is regarded as scheduling unit generally. Each patient has a different route of surgery, that is, the surgical stages of the patients are different. At each stage, each patient selects any set of available surgical resources from the set of candidate surgical resources. Therefore, through the analysis of the scheduling problem of surgical cases, this study model this scheduling as FJSP.

3.1 Assumptions

- All the patients are ready before the surgery.
- Different surgeries have the same priority.
- A surgical resource set can handle only one surgical stage at a time.
- Once a surgery is performed, it cannot be interrupted until it is completed.
- The patient will move to the next stage only when the previous surgical stage is completed.
- The state of every kind of surgical resource set is known before scheduling begins.
- The processing time of each surgical stage is determined in advance and will not change with the order.
- The time and cost of the preparation and switching process are considered.

3.2 Notations

Indices	
i, i'	Indices of patients
l, l'	Indices of surgical stages
k, k', k''	Indices of surgical resource sets
Parameters	
n	Number of the patients
m	Number of surgical resource sets
op_i	Number of surgical stages of patient i
$O_{i,l}$	lth surgical stage of patient i
U	Large positive number
Variables	
$PT_{i,l,k}$	Processing time of $O_{i,l}$ on the kth surgical resource set
$ST_{i,k',k}$	Switching time of patient i from the k'th surgical resource set to the kth surgical resource set

$PP_{i,l,k}$	Preparation time of $O_{i,l}$ on the kth surgical resource set
$PC_{i,l,k}$	Unit time medical cost of $O_{i,l}$ on the kth surgical resource set
$SC_{i,k',k}$	Unit time medical cost of the switching process for patient i from the k'th surgical resource set to the kth surgical resource set
$PPC_{i,l,k}$	Unit time medical cost of the preparation process of $O_{i,l}$ on the kth surgical resource set
SM	Medical cost of patients during surgery
PPM	Medical cost of preparation process
SPM	Medical cost of switching process
TMC	Total medical cost
$CM_{i,l}$	Completion time of $O_{i,l}$
$EB_{i,l,k}$	Binary variable taking value 1 if $O_{i,l}$ can be processed by kth surgical resource set, and 0 otherwise
$X_{k,i',l',i,l}$	Binary variable taking value 1 if on the kth surgical resource set, $O_{i,l}$ is processed after $O_{i',l'}$, and 0 otherwise
$Y_{i,l,k,k'}$	Binary variable taking value 1 if $O_{i,l}$ is processed by the kth surgical resource set and $O_{i,l-1}$ is processed by the k'th surgical resource set

3.3 Problem formulation

Aiming at the surgical case scheduling problem, a sequence-based MILP model is established to minimize the completion time (*Makespan*) of patients and the total medical cost (*TMC*). The model includes objective functions and constraints. To establish the objective function, the total medical cost modules of the surgery are also provided.

The medical costs vary in accordance with the surgical stage. Consequently, the *TMC* consists primarily of several parts: the medical cost during the surgery (*SM*), the medical cost in the patient switching process (*SPM*), the medical cost in the patient preparation process (*PPM*).

SM can be calculated by determining the surgical resource set k occupied by the patient and multiplying the unit time medical cost and processing time of $O_{i,l}$ on the kth surgical resource set, as shown in Eq. (1)

$$SM = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{op_i}\sum_{k'=1}^{m} Y_{i,l,k,k'} \cdot PC_{i,l,k} \cdot PT_{i,l,k} \quad (1)$$

SPM can be calculated by judging the surgical resource set k occupied by the patient and multiplying the unit time medical cost of the switching process and the switching time of the patient i from the k'th surgical resource set to the kth surgical resource set, as expressed in Eq. (2)

$$SPM = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{op_i}\sum_{k'=1}^{m} Y_{i,l,k,k'} \cdot SC_{i,k',k} \cdot ST_{i,k',k} \quad (2)$$

PPM can be calculated by judging the surgical resource set k occupied by the patient and multiplying the unit time medical cost of the preparation process and the preparation time of $O_{i,l}$ on the kth surgical resource set, *PPM* can be calculated by Eq. (3)

$$PPM = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{op_i}\sum_{k'=1}^{m}\sum_{i'=1}^{n} Y_{i,l,k,k'} \cdot X_{k,i',l',i,l} \cdot PPC_{i,l,k} \cdot PP_{i,l,k} \quad (3)$$

TMC consists of *SM*, *SPM*, and *PPM*. Therefore, the *TMC* can be calculated by Eq. (4)

$$TMC = SM + SPM + PPM \quad (4)$$

According to the problem description, the scheduling optimization objectives are to minimize the *Makespan* and the *TMC*. Therefore, the model is formulated as follows.

$$\min C_{max} \quad (5)$$

$$\min TMC \quad (6)$$

Subject to

$$\sum_{k=1}^{m}\sum_{k'=1}^{m} Y_{i,l,k,k'} = 1\ \forall i, l \quad (7)$$

$$\sum_{k=1}^{m} Y_{i,1,k,0} = 1\ \forall i \quad (8)$$

$$\sum_{k'=1}^{m} Y_{i,l,k,k'} \le EB_{i,l,k}\ \forall i, l, k \quad (9)$$

$$Y_{i,1,k,0} \le EB_{j,1,k}\ \forall i, k \quad (10)$$

$$Y_{i,l,k,k'} \le \sum_{k''=1}^{m} Y_{i,l-1,k',k''}\ \forall i, l > 2, k, k' \quad (11)$$

$$Y_{i,2,k,k'} \le Y_{i,1,k',0}\ \forall i, k, k' \quad (12)$$

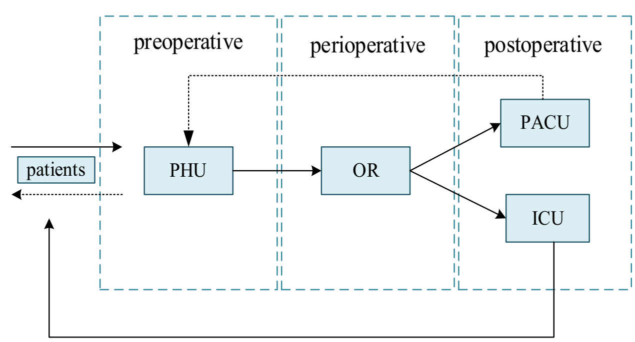


Fig. 1 Surgery procedure with different stages

$$CM_{i,l} \geq CM_{i,l-1} + \sum_{k=1}^{m}\sum_{k'=1}^{m} Y_{i,l,k,k'} \cdot \left(PT_{i,l,k} + ST_{i,k',k} + PP_{i,l,k}\right) \quad \forall i,\, l > 1,\, l' > 1 \tag{13}$$

$$CM_{i,1} \geq \sum_{k=1}^{m} Y_{i,1,k,0} \cdot \left(PT_{i,1,k} + ST_{i,0,k} + PP_{i,1,k}\right) \forall i \tag{14}$$

$$CM_{i,l} \geq CM_{i',l'} + PT_{i,l,k} + PP_{i,l,k} - U \cdot \left(3 - X_{k,i,l,i',l'} - \sum_{k'=1}^{m} Y_{i,l,k,k'} - \sum_{k'=1}^{m} Y_{i'l',k,k'}\right) \quad \forall i, l > 1, i' \neq i, l' > 1, k \tag{15}$$

$$CM_{i,1} \geq CM_{i',l'} + PT_{i,1,k} + PP_{i,l,k} - U \cdot \left(3 - X_{k,i,1,i',l'} - \sum_{k'=1}^{m} Y_{i,1,k,0} - \sum_{k'=1}^{m} Y_{i'l',k,k'}\right) \quad \forall i, l > 1, i' \neq i, l' > 1, k \tag{16}$$

$$CM_{i,l} \geq CM_{i',1} + PT_{i,l,k} + PP_{i,l,k} - U \cdot \left(3 - X_{k,i,l,i',l'} - \sum_{k'=1}^{m} Y_{i,l,k,k'} - \sum_{k'=1}^{m} Y_{i',1,k,0}\right) \quad \forall i, l > 1, i' \neq i, l' > 1, k \tag{17}$$

$$CM_{i,1} \geq CM_{i',1} + PT_{i,1,k} + PP_{i,l,k} - U \cdot \left(3 - X_{k,i,1,i',1} - \sum_{k'=1}^{m} Y_{i,1,k,0} - Y_{i',1,k,0}\right) \quad \forall i, l > 1, i' \neq i, k \tag{18}$$

$$CM_{i',l'} \geq CM_{i,l} + PT_{i',l',k} + PP_{i,l,k} - U \cdot \left(X_{k,i,l,i',j'} - \sum_{k'=1}^{m} Y_{i,l,k,k'} - \sum_{k'=1}^{m} Y_{i',l',k,k'} + 2\right) \quad \forall i, l > 1, i' \neq i, k, l' > 1 \tag{19}$$

$$CM_{i',l'} \geq CM_{i,1} + PT_{i',l',k} + PP_{i,l,k} - U \cdot \left(X_{k,i,1,i',l'} - Y_{i,1,k,0} - \sum_{k'=1}^{m} Y_{i',l',k,k'} + 2\right) \quad \forall i, l > 1, i' \neq i, k, l' > 1 \tag{20}$$

$$CM_{i',1} \geq CM_{i,l} + PT_{i',1,k} + PP_{i,l,k} - U \cdot \left(X_{k,i,l,i',1} - \sum_{k'=1}^{m} Y_{i,l,k,k'} - Y_{i',1,k,0} + 2\right) \quad \forall i, l > 1, i' \neq i, k \tag{21}$$

$$CM_{i',1} \geq CM_{i,1} + PT_{i',1,k} + PP_{i,l,k} - U \cdot \left(X_{k,i,1,i',1} - \sum_{k'=1}^{m} Y_{i,1,k,0} - Y_{i',1,k,0} + 2\right) \quad \forall i, i' \neq i, k \tag{22}$$

$$C_{\max} \geq C_{i,op_i} \forall i \tag{23}$$

$$C_{i,op_i} \geq 0 \forall i \tag{24}$$

$$PC_{i,l,k} \geq 0 \forall i, l, k \tag{25}$$

$$SC_{i,k',k} \geq 0 \forall i, k, k' \tag{26}$$

$$PPC_{i,l,k} \geq 0 \forall i, i', k \tag{27}$$

$$X_{k,i',l',i,l}, Y_{i,l,k,k'} \geq 0 \tag{28}$$

Constraints (5) and (6) are objective functions. Constraints (7) and (8) consider that each surgical stage is allocated to a surgical resource set for processing. For each surgical stage, there is a set of available surgical resource sets to process. Therefore, constraints (9) and (10) are defined to ensure the surgical resource set occupied by each surgical stage is selected from the available surgical resource sets. According to the decision variables $Y_{i,j,k,k'}$, $O_{i,l\text{-}1}$ can be processed by the k'th surgical resource set, if $O_{i,l}$ can be processed by the kth surgical resource set. Constraints (11) and (12) guarantee that $O_{i,\ l\text{-}1}$ is processed by the k'th surgical resource set, $O_{i,l}$ is processed by the kth surgical resource set. Constraints (13) and (14) ensure that the patients will not enter the next surgical stage until the previous stage has been completed. Constraints (15)–(22) ensure that each surgical resource set is occupied by only one surgical stage at a time. Constraint (23) determines the value of the *Makespan* by considering the completion time of the last stage of all the patients. Constraints (24)–(28) force these variables to be positive.

4 The proposed algorithm

The imperialist competitive algorithm (ICA) is a new metaheuristic inspired by sociopolitical behaviors. In contrast to other optimization algorithms such as the GA and PSO, the ICA has good neighborhood search ability, effective global search properties, and a good convergence rate [32, 64, 65]. However, the ICA cannot simultaneously handle objective conflicts in multi-objective design problems. The ICA has the disadvantage of capturing local optimal solutions when used for high-dimensional or complex multimodal functions [37, 66]. The surgical case scheduling problem in our study considers *Makespan* and *TMC* as two objectives. To better solve the research problems with ICA, an improved multi-objective ICA, named IMOICA, is proposed. The main features of the IMOICA can be described as following: (1) the social hierarchy strategy is developed to initialize the empire; (2) to enhance the global search ability of the algorithm, the concept of AR is introduced into the assimilation strategy; (3) the revolution strategy is utilized for the diversity of the population; (4) the VNS strategy is used to improve the exploitation capacity of the algorithm. The framework of the IMOICA is described in Algorithm 1.

Algorithm 1. IMOICA
Input: all populations
Output: Pareto optimal solution

```
Randomly produce the initial population
Initialize the empires (c.f. Section 4.2)
For each imperialist i do
  | Generation of Pareto solution (c.f. Section 4.3)
  | For each colony j do
  |   | Perform the assimilation strategy with AR measure (c.f. Section 4.4)
  |   | Conduct the updating strategy of empire (c.f. Section 4.5)
  |   | Implement the revolution strategy (c.f. Section 4.6)
  |   | Execute the VNS strategy (c.f. Section 4.7)
  | End
End
```

In the course of the study, the country is equivalent to the individual, all countries (individuals) constitute the population, and optimizing the country (individual) is equivalent to optimizing the surgical case scheduling of a group of patients. The scheduling sequence is obtained by optimization algorithm. The application of the algorithm in surgical case scheduling is shown in Fig. 2.

4.1 Encoding

For the n patients who need to undergo surgery, the encoding is based on the sequence of different stages of the surgery and the different resources selected. The encoding solution consists of two parts: the surgical resource assignment (SRA), and the surgical stage sequence (SSS). Figure 3 shows the composition of an encoding solution. The SRA stores the available surgical resource set numbers selected for each surgical stage. The SSS defines the patient numbers. Furthermore, the Local Selection strategy [67] is adopted in the SRA part and the random selection strategy is adopted in the SSS part.

4.2 Initialization of empires

The social hierarchy strategy is used to initialize the empire. First, all countries are sorted according to fitness. As shown in Fig. 4, the best fitness (surgical case scheduling) with the highest social hierarchy is named α imperialist. In succession, the following *Nim*-1 levels are β imperialist, γ imperialist, and so on. The remaining countries are colonies. The colonies are divided according to the social hierarchy of the imperialists, i.e., the higher the social hierarchy, the more colonies will be occupied. Note that the occupation of colonies by imperialists is random. Therefore, after the selection of the *Nim* imperialists, the order of the remaining colonies will be disrupted. Then, the imperialists and their colonies form different empires. The number of colonies occupied by imperialists is obtained by Eq. (29).

$$Num(x) = \frac{Ncl}{(x+1)^2} \tag{29}$$

where x = 1, …, *Nim*; *Nim* is the number of imperialists, *Num*(x) is the number of colonies acquired by x imperialist, and *Ncl* is the number of colonies. During the initialization of the empire, the smaller the x, the higher the social hierarchy of the empire.

Algorithm 2. Non-dominant sorting

Ω = find_nondominated_front (P);
For each $p' \in P \wedge p' \notin \Omega$ **do**
  $\Omega = \Omega \cup \{p'\}$
  For each $q \in \Omega \wedge q \neq p'$ **do**
    If $p' \succ q$ **do**
      $\Omega = \Omega \setminus \{q\}$
    End
    Else $p' \prec q$ **do**
      $\Omega = \Omega \setminus \{p'\}$
    End
  End
End

4.3 Generation of Pareto solution

By using the sorting non-dominant strategy [68], the optimal solution in the population is selected as the Pareto solution. The initial population and the construction set are denoted by P and Ω, respectively. The individuals in Ω are temporary, because they can be deleted in subsequent comparisons. At the beginning of the algorithm, the first individual is inserted into the construction set Ω, and the individual p' in the evolutionary population P ($p' \notin \Omega$) is removed and inserted into the construction set Ω. Then, the individual p' is compared with the individual in Ω in turn, and the individual dominated by p' is deleted. Assuming that p' is dominated by any individual in Ω, p' is deleted

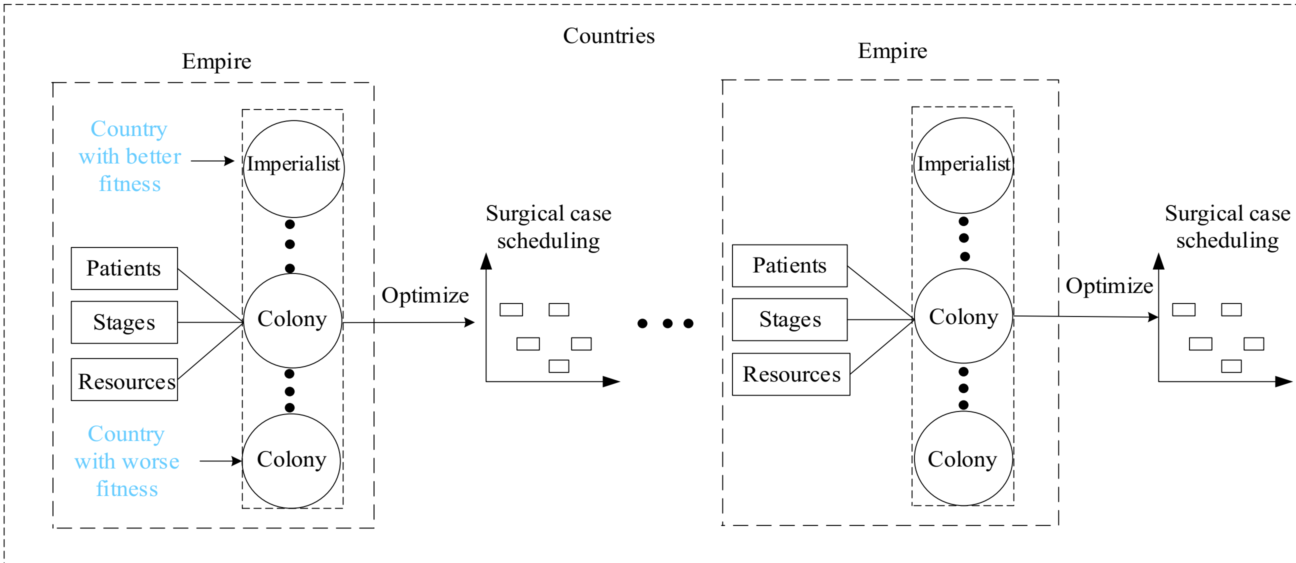


Fig. 2 Application of the algorithm in surgical case scheduling

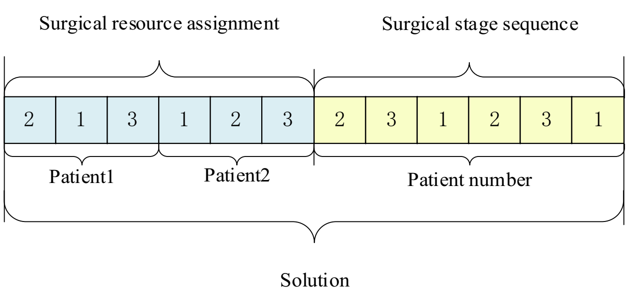


Fig. 3 Example of an encoding solution

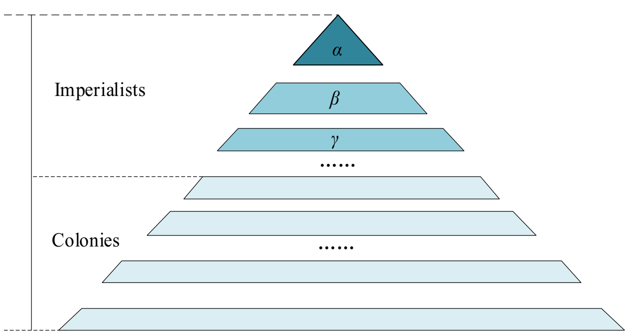


Fig. 4 Social hierarchy divided by the countries

from Ω. The framework for the non-dominant sorting method is described in Algorithm 2.

In the first iteration, there are Nim empires, each imperialist i has Ncli colonies. Hence, the strategy begins with those Nim empires. For each empire, the imperialist is preserved to form a Pareto solution. Then, the imperialist is compared with the remaining colonies. The imperialist remains in the Pareto solution set as long as the imperialist dominates the colonies; otherwise, the imperialist is replaced by the colony. In the following iteration, the imperialists of the empires are compared with the individual solutions of the previous Pareto solution, and the Pareto optimal solution is obtained. This strategy is shown in Fig. 5.

4.4 Assimilation strategy with AR measure

The important factor affecting the assimilation strategy is the distance between imperialists and colonies. The idea is to use the AR measure based on this distance to improve the performance of the algorithm and achieve the globally optimal position. The average distance between the imperialist and its colonies (AVR) is as follows:

$$AVR = \frac{1}{M}\sum_{i=1}^{M}\left[\left(V_{best}^{col} - V^{Imp}\right) - \left(V_{current}^{col} - V^{Imp}\right)\right] \quad (30)$$

where M is the number of colonies in each empire, V_{best}^{col} is the fitness of the best colony, $V_{current}^{col}$ is the fitness of the current colony, and V^{Imp} is the fitness of the imperialist.

After calculating the value of AVR, the following three scenarios occur, in accordance with the value of AVR:

- If the value of AVR is less than a specified threshold (ω) that is set in advance, the mutation operator is performed. Figure 6a shows the mutation operator that occurs in the SRA part. First, a feasible individual (P_0) is selected. Then, several elements in the SRA part are chosen randomly and replaced with available surgical resource set numbers. The new individual (P_0') is generated. Figure 6b presents the mutation operator that acts on the SSS part. First, three variation points are randomly selected in the individual (G_0) in accordance

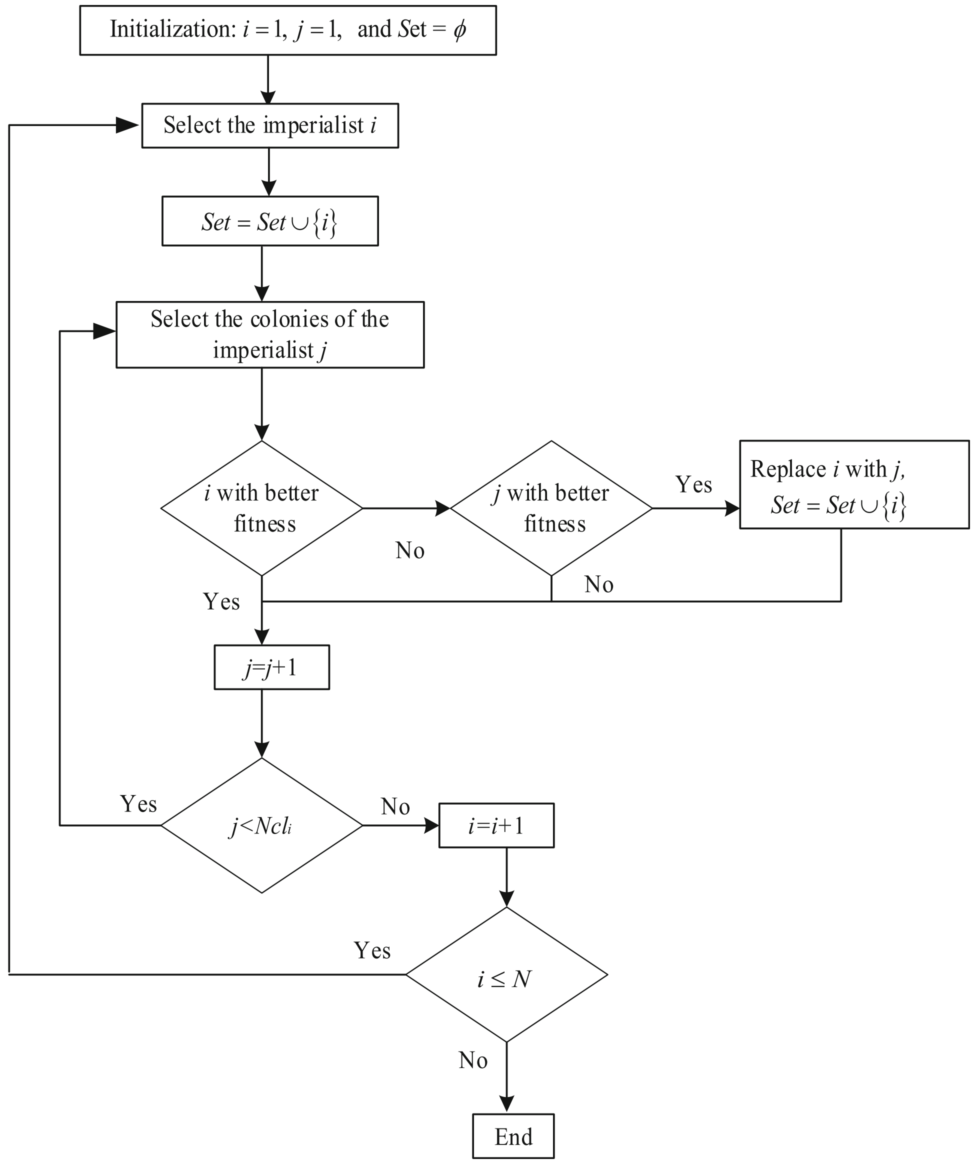


Fig. 5 Generation of Pareto solutions

with the generated neighborhood search range. Then, all combinations of neighborhood solutions are generated (G_0', G_0''). Finally, one of the neighborhood solutions is selected randomly as the offspring.

- If the value of AVR is greater than ω, the crossover operator is performed. Figure 7a is the two-point crossover strategy that acts on the SRA part. First, two positions of the individual (P_1) are selected randomly. Then, selecting another individual (P_2), the elements between these two positions of P_1 are selected and inserted into P_2 to form a new individual (P_3). Figure 7b shows the JBX crossover strategy that acts on the SSS part. First, the patient numbers are randomly selected from individual (Q_1), and according to the position, the patient numbers are inserted into the offspring (Q_2). Then, another individual (Q_3) is selected, and the remaining patient numbers are inserted into Q_2 in turn.
- If the value of AVR is equal to ω, then the classical assimilation strategy is conducted. The assimilation process can be achieved through the movement of colonies to imperialists. The movement of a colony to the imperialist is presented in Fig. 8.

The colony moves to the imperialist through a distance of X to a new position, and X is defined as

$$X \sim U(0\beta \times d) \tag{31}$$

where β is a real number greater than 1, and d is the distance between the colony and the imperialist. To find the difference points around the imperialists and expand the scope of search, a random deviation from direction θ is

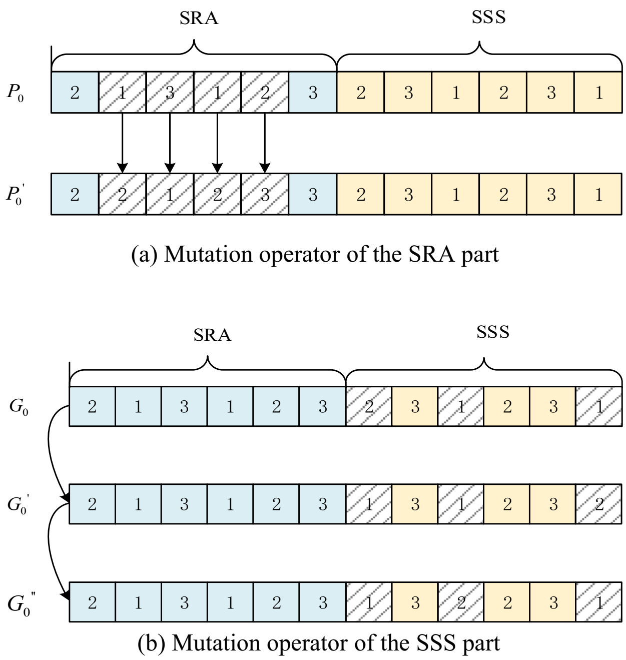


Fig. 6 Mutation operator

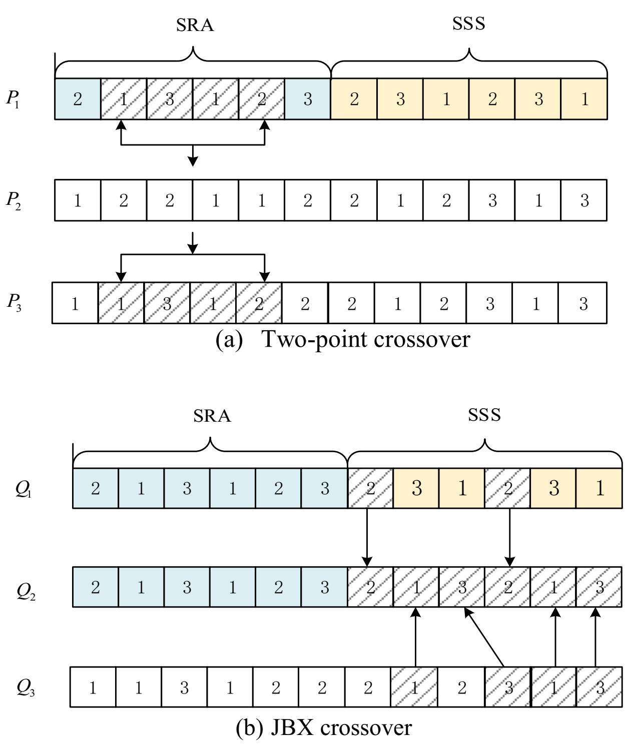


Fig. 7 Crossover operator

added. θ is subject to a uniformly distributed random number:

$$\theta \sim U(-\gamma\gamma) \tag{32}$$

where γ represents a parameter used to adjust deviation from the original direction.

The framework for the assimilation strategy with AR measure is described in Algorithm 3.

Algorithm 3. Assimilation strategy with AR measure

Calculate the value of AVR
Select threshold ω
IF $AVR < \omega$ **do**
  The mutation operator is performed
End
IF $AVR > \omega$ **do**
  The crossover operator is conducted
End
Else
  The classical assimilation strategy is implemented
End

Algorithm 4. Updating of the empire

Execute the first update strategy
For each empire i **do**
  IF the colony in empire i is better than its imperialist **do**
    Replace the original imperialist as a new imperialist
  End
End
Execute the second update strategy
For each empire i **do**
  IF there are no colonies in the empire i **do**
    The strategy of empire extinction is implemented, that is to say, the imperialist is transferred to the powerful empire and become one of its colonies.
  End
End

4.5 Updating of the empire

The updating of the empire includes two situations. (1) Replacement of imperialist: with the increase of iterations, the colonies might obtain more power than their imperialist in an empire. Consequently, the empire is replaced, i.e., the most powerful colony will replace the imperialist as the new imperialist, as depicted in Fig. 9a. (2) Extinction of empire: for the empires that do not have any colonies, the empire extinction strategy is implemented, i.e., for the empires that exist only one imperialist will be transferred to the powerful empire and become one of its colonies. Figure 9b shows another situation in the updating of the

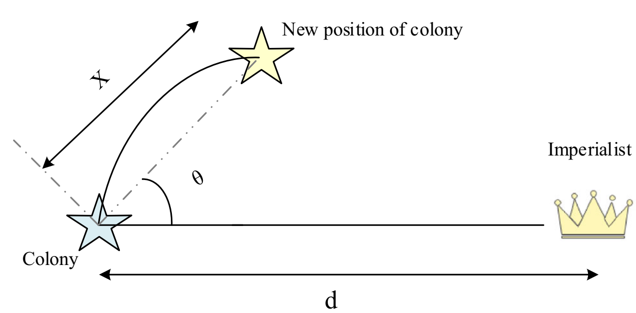


Fig. 8 Movement of the colonies toward the imperialist

empire. The framework for the updating of the empire is presented in Algorithm 4.

4.6 Revolution strategy

Revolution is another way to generate new solutions. The detailed steps of the revolution strategy are list in Algorithm 5, where R and α are integers, Pr is the revolution probability (consistent with the parameter in reference [70]), S_i is the number of countries in empire i, and *rand* is a randomly generated number.

For colony λ, U_λ represents the number of colonies that dominate the colony λ in the empire. Good colonies have a greater possibility of carrying out the revolution strategy. Therefore, assuming that the colony λ with the smallest U_λ value, it will have more opportunity to execute the revolution strategy to produce a new colony.

The change and insert operators are used in the revolution strategy. The change operator produces a new solution by replacing an element randomly in the SRA part, as shown in Fig. 10a. The insert operator consists of selecting two position elements r_1 and r_2, randomly in the SSS part. The element r_2 is inserted into the r_1, and the element r_1 is inserted into the last position. Then, the element after r_2 moves forward. Figure 10b shows an example of the insert operator.

Algorithm 5. Revolution

```
For i = 1 to N_im do
    α ← 1
    For j = 1 to S_i do
        If rand < P_r do
            α=α+1
        End
    End
    Change the Pareto front by comparing all the colonies in the empire.
    If α > 1 do
        For β =1 to R do
            Select a colony λ with the least U_λ
            If r = 1 do
                The insert operator is conducted in colony λ
            End
            Else
                Perform the change operator in the colony λ
            End
            A new solution SL0 is obtained
            IF SL0 dominates λ do
                The colony is replaced
                For all individuals in the Pareto front do
                    If z dominates m do
                        Pareto front is updated
                    End
                End
            End
        End
    End
End
```

4.7 Variable neighborhood search

VNS is a heuristic algorithm to solve the optimization problem. VNS has the characteristics of simple structure and few parameters, and its unique variable neighborhood mechanism can prevent the search from falling into local optimization [69]. This provides VNS with strong local search capability [70].

In this paper, the VNS strategy is added to enhance the convergence performance of the algorithm. The steps of VNS are listed in Algorithm 6, in which three neighborhood structures: insert, change, and swap are used. The change and insert operators are as described in Sect. 4.6. The swap operator acts on the SSS part, selecting two position elements randomly and then swapping them. *max_0* is the maximum number of cycles the objective function evaluates for termination.

Infeasible solutions are dropped. The generated feasible solution is compared with the current solution, and if the feasible solution is better, the current solution is replaced.

Algorithm 6. Framework of VNS

```
Select the first member of the Pareto front as the current solution x
  v, g ← 1
  While v ≤ max_0 do
      If g < 4 do
          r = rand () %3
          If r = 0 do
              Conduct the swap operator
          End
          Else if r = 1 do
              Conduct the insert operator
          End
          Else
              Conduct the change operator
          End
          Generate a new solution z
          If z dominates x do
              Update the Pareto front
              Let g ← 1
          End
          Else
              g ← g+1
          End
          v ← v+1
          If v is divisible by integer Ln do
              Randomly choose a solution y and instead of solution x
          End
      End
  End
```

4.8 Algorithm complexity

In this part, the complexity of the main strategy of the algorithm is presented. In the process of generation of Pareto solution, the complexity of the algorithm is $O(p'q)$, p' represents the number of individuals in the population, q represents the number of individuals in the Pareto solution sets. In assimilation strategy with AR measure, the complexity of the algorithm is $O(2mns)$, m represents the

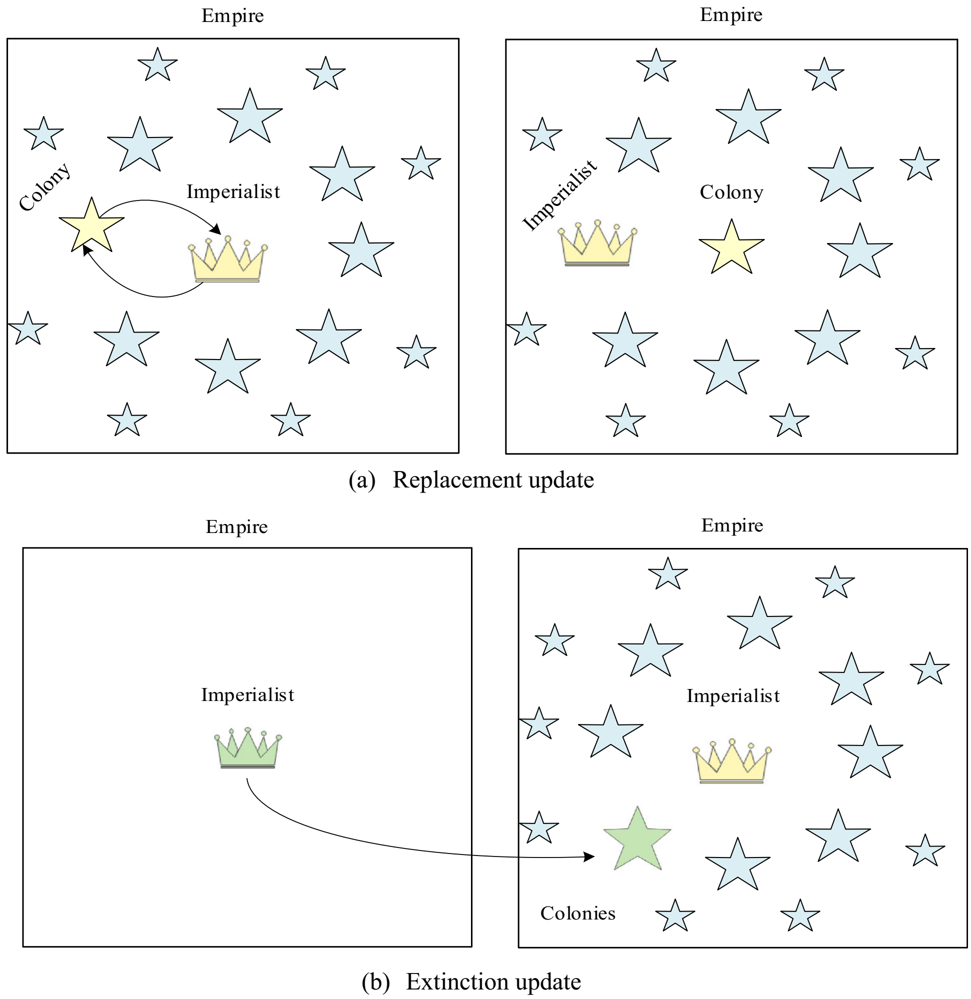


Fig. 9 Updating of the empire

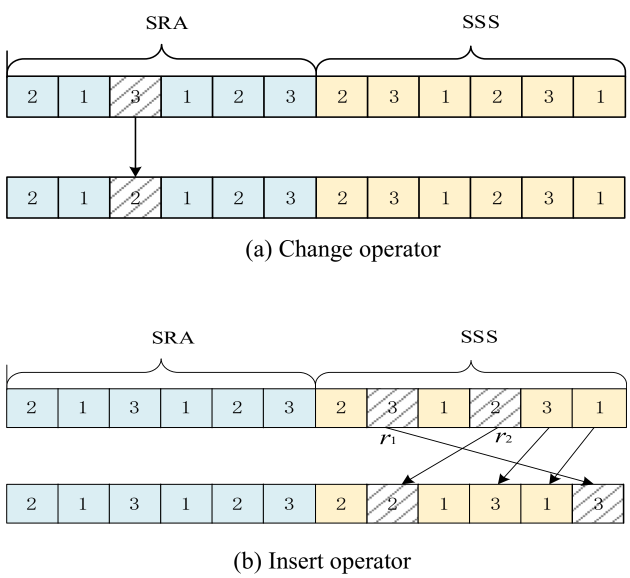


Fig. 10 Change and Insert operators in the revolution strategy

number of empires, n denotes the number of colonies, and s is the length of the surgical resource assignment or the surgical stage sequence. The complexity of updating of the empire algorithm is $O(mn)$. The complexity of revolution strategy is $O(m(f + r))$, f represents the number of colonies in each empire, r represents a positive integer less than 10. In the variable neighborhood search strategy, the complexity of the algorithm is $O(logv)$, v is the maximum number of cycles the objective function evaluates for termination.

5 Experimental results

The performance of the algorithm was evaluated by simulating the surgical case scheduling process. The characteristics of systems and resources were briefly described in Table 2. Before the beginning of the experiment, we investigated a third-class hospital, analyzed and collated the data obtained, and then generated an experimental instance. The generated instances were processed by the proposed algorithm and the scheduling scheme was generated. In addition, in the practical application part, the simulation experiment was carried out by using the data disclosed in other papers to verify the algorithm.

First, the Taguchi method was used to investigate the influence of important parameters on the algorithm. Then, the model was verified using CPLEX. The strategies were verified through numerical experiments. Finally, to assess the performance of the algorithm, the proposed algorithm was compared with other recently published efficient algorithms.

5.1 Experimental parameters

The experimental parameters included the population size (P), the number of imperialists (Nim), a specific threshold (ω), the integer (R, Ln), and the maximum number of cycles the objective function evaluates to terminate (max_0). The levels of the six parameters were presented in Table 3. The Taguchi method was used to examine the influence of these six parameters on the performance of the proposed algorithm. Table 4 shows the orthogonal array L_{27} (3^6) in which each combination runs 30 times in 30 s, and the means of *Makespan* and *TMC* are used as the response variable (RV). Figure 11 reports the factor level trend of the six parameters, from which it can be seen that, when $P = 150$, $Nim = 15$, $\omega = 5$, $R = 20$, $max\_0 = 50$, and $Ln = 3$, the IMOICA has the best performance.

5.2 Comparison with the exact CPLEX solver

The CPLEX solver can be used to verify the accuracy of the model and evaluate the performance of the proposed algorithm further. The solver uses the exact algorithm based on branch and bound, consuming substantial time to calculate accurate results. Consequently, when comparing the CPLEX algorithm with the proposed algorithm, the computing time for CPLEX was set to 3 h, the number of threads was set to 3, and the CPU stop time of IMOICA was set to 30 s. Then, 9 small-scale instances were tested.

Table 3 The level of the key parameter

Parameter	Level		
	1	2	3
P	50	100	150
Nim	5	8	15
ω	5	10	20
R	8	10	20
max_0	10	20	50
Ln	3	5	8

To conduct a comprehensive comparison between CPLEX and IMOICA, the inverted generational distance and hypervolume are used as comprehensive indicators to evaluate the convergence and distribution of the algorithms.

Inverted generational distance (IGD) [71]: Average distance from the real and uniformly distributed Pareto optimal solution set Q * to the optimal solution set Q obtained by the algorithm.

$$IGD(Q, Q^{*}) = \frac{1}{|Q^{*}|} \sum_{v \in Q^{*}} \min_{z \in Q} d(v, z) \tag{33}$$

where $d(v, z)$ corresponds to the distance between solutions v and z. $|Q^{*}|$ is the number of solutions of the point set distributed on the real Pareto front. A smaller value of IGD can be considered as indicating a better set of solutions approximating the true Pareto front from the convergence.

Hypervolume (HV) [71]: Volume of the region in the target space surrounded by the non-dominant solution set and reference points obtained by the algorithm.

$$HV(Q) = Leb\left(\underset{z \in Q}{\cup} [f_1(x), r_1] \times \ldots [f_d(x), r_d] \right) \tag{34}$$

where $r = (r_1, r_2, \ldots, r_d)$ is the reference point, Q is the non-dominant solution of the algorithm, d is the number of objective functions, and $Leb(\cdot)$ is the Lebesgue measure. A larger HV value indicates a better performance.

Table 5 lists the IGD and HV values achieved by the two algorithms, where the best IGD and HV values are marked

Table 2 The characteristics of systems and resources

Computer		Software		
		Visual studio	MATLAB	CPLEX
Version	Windows 10	16.6.2 visual Studio Community 2019	9.0.0.341360 (R2016a)	12.7.1.0 IBM ILOG CPLEX Optimization Studio
Processor	Intel(R) Core (TM) i7-6700 CPU @ 3.40 GHz 3.41 GHz	-	-	-
System type	64bit	64bit	64bit	64bit

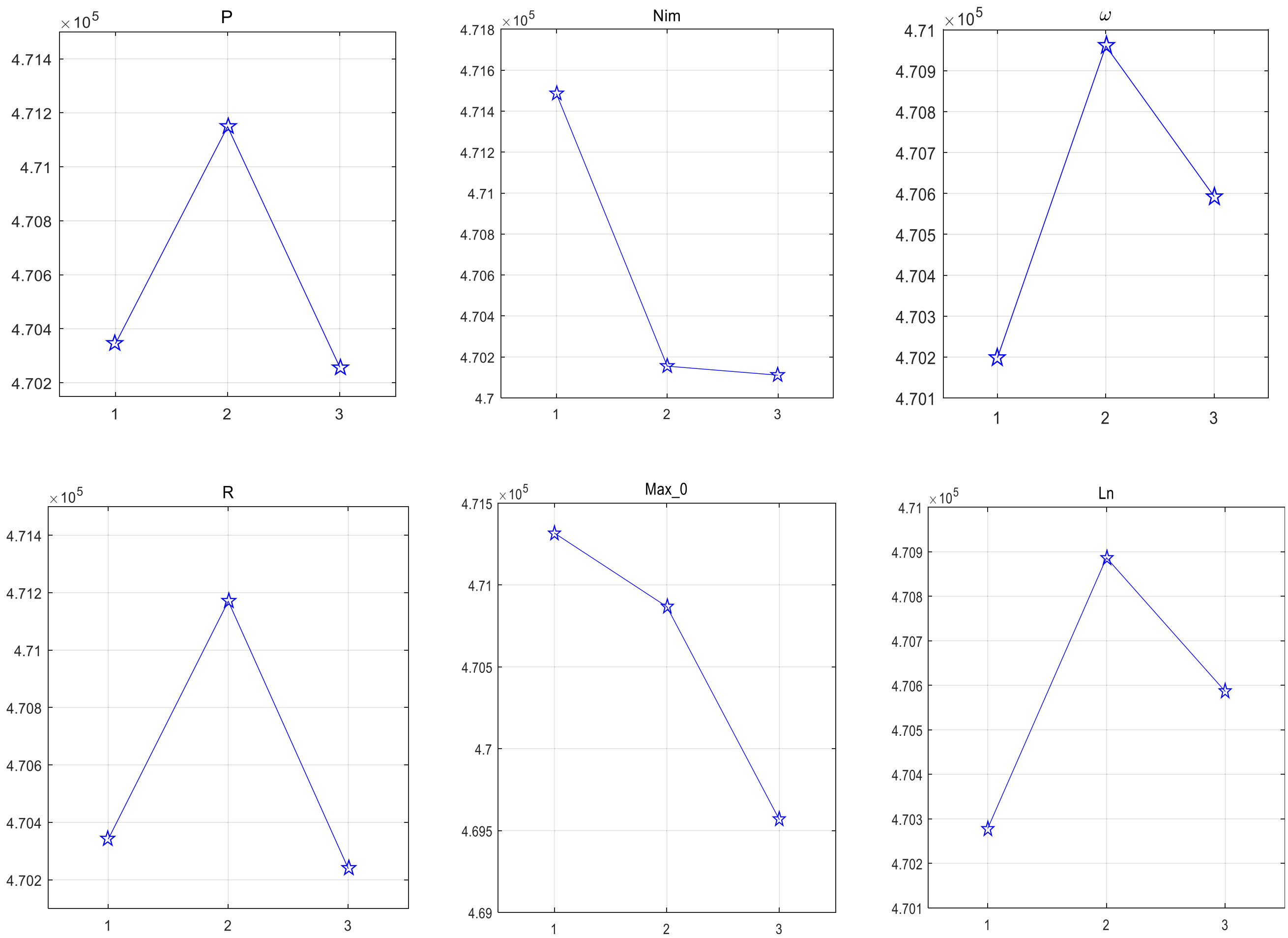


Fig. 11 Factor level trends of key parameters

in bold. Table 5 contains six columns. The first column presents the instance name and the second column shows the surgical case. The following two columns correspond to the IGD values obtained by IMOICA and CPLEX. The last two columns list the HV values obtained by IMOICA and CPLEX. The bold in the Table is the optimal values obtained by comparing different algorithms under different instances. According to the results, it can be seen that the proposed algorithm outperforms CPLEX in terms of the convergence and distribution of the algorithm.

5.3 Efficiency of the assimilation strategy with AR measure

To investigate the performance of the assimilation strategy with AR measure, which was discussed in Sect. 4.4, two types of assimilation strategy were studied: (1) the proposed assimilation strategy with AR measure; and (2) the assimilation strategy adopted by Zandieh et al. [72] (called IMOICA_NAR). First, 21 different surgical cases base on practical problems were generated and used for simulation tests. The two algorithms were performed under the same operating environment and each instance was run 30 s. To verify the effectiveness of the proposed algorithm, the relative percentage increase (RPI) [21] is introduced for the multifactor analysis of variance (ANOVA) comparison.

Table 6 (where *j-m* represents *j* patients and *m* surgical resource sets) lists the IGD and HV values achieved by the two algorithms. It can be found in Table 6: (1) for the IGD values, IMOICA obtains 20 better results; (2) for the HV values, 19 better results are obtained by the IMOICA, compared with IMOICA_NAR algorithm; and (3) the ANOVA results from Fig. 12 illustrates that the IMOICA is better than its comparison algorithm.

5.4 Efficiency of the revolution strategy

To verify the effectiveness of the revolution strategy, we compared IMOICA with IMOICA_NRS, IMOICA_NRS without the proposed revolution strategy. The two algorithms were performed under the same operating environment with each instance running 30 s. The comparison of

Table 4 Combination of algorithm parameters

Number	Factor						RV
	P	*Nim*	ω	*R*	*max_0*	*Ln*	
1	1	1	1	1	1	1	4.725×10^5
2	1	1	1	1	2	2	4.709×10^5
3	1	1	1	1	3	3	4.685×10^5
4	1	2	2	2	1	1	4.695×10^5
5	1	2	2	2	2	2	4.699×10^5
6	1	2	2	2	3	3	4.733×10^5
7	1	3	3	3	1	1	4.692×10^5
8	1	3	3	3	2	2	4.699×10^5
9	1	3	3	3	3	3	4.694×10^5
10	2	1	2	3	1	2	4.738×10^5
11	2	1	2	3	2	3	4.746×10^5
12	2	1	2	3	3	1	4.678×10^5
13	2	2	3	1	1	2	4.739×10^5
14	2	2	3	1	2	3	4.695×10^5
15	2	2	3	1	3	1	4.680×10^5
16	2	3	1	2	1	2	4.721×10^5
17	2	3	1	2	2	3	4.690×10^5
18	2	3	1	2	3	1	4.715×10^5
19	3	1	3	2	1	3	4.724×10^5
20	3	1	3	2	2	1	4.700×10^5
21	3	1	3	2	3	2	4728×10^5
22	3	2	1	3	1	3	4.690×10^5
23	3	2	1	3	2	1	4.714×10^5
24	3	2	1	3	3	2	4.669×10^5
25	3	3	2	1	1	3	4.694×10^5
26	3	3	2	1	2	1	4.726×10^5
27	3	3	2	1	3	2	4.678×10^5

IGD and HV values of the two algorithms was shown in Table 7. According to Table 7: (1) for the IGD values, IMOICA obtains 21 better results; (2) for the HV values, 18 better results are obtained by the IMOICA, compared with IMOICA_NRS algorithm; and (3) the ANOVA results from Fig. 13 illustrates that the IMOICA is better than its comparison algorithm. The main reason for this effect is that by using the revolution strategy can produce new solutions and further increase the diversity of the population.

5.5 Efficiency of the VNS strategy

In this section, we compared IMOICA with IMOICA_NVS, IMOICA_NVS without the proposed VNS strategy. The two algorithms were performed under the same operating environment with each instance running 30 s. Table 8 lists the IGD and HV values achieved by the two algorithms. As can be seen from the comparison results in Table 8: (1) for the IGD values, IMOICA obtains 19 better results; (2) for the HV values, 19 better results are obtained by the IMOICA, compared with IMOICA_NVS algorithm; and (3) the ANOVA results from Fig. 14 shows that the IMOICA is better than its comparison algorithm. The main reason for this effect is that by using the VNS strategy can increase the search range and effectively prevent the algorithm from falling into local optimization.

5.6 Comparison with other efficient algorithms

We compared IMOICA with current popular algorithms similar to ours, including the Pareto-based grouping discrete harmony search (PGDHS) algorithm [48], the hybrid evolutionary algorithm based on decomposition (HMOEA/D) [52], the Pareto-based discrete artificial bee colony algorithm (P-DABC) [73], and the non-dominated sorted genetic algorithm (NSGA-II) [74]. The algorithms adopted the parameter settings proposed in their references. For fair comparison, all the algorithms ran in the same environment for 30 s and got the Pareto solutions. Figure 15 presents the Pareto values of the two different scale instances. It should be noted that there is none true Pareto front, because the generated 21 instances are generated by a realistic surgical case scheduling system. Therefore, we used the Pareto solutions obtained by the five compared algorithms as the near Pareto front "PF". It can be seen from the figure that the IMOICA proposed in this paper can get more and better Pareto values.

Table 9 lists the comparison results of the IGD values of the given 21 instances. The first column gives the scale of the instance, while the second column displays the best IGD values collected from the five compared algorithms. Then, the IGD values collected by IMOICA, PGDHS, HMOEA/D, P-DABC, and NSGA-II, are displayed in the following five columns, respectively, while the RPI values obtained from the five compared algorithms are provided in the last five columns. According to Table 9: (1) in comparison with the other four algorithms, the proposed IMOICA obtained 19 optimal RPI values for the 21 specified instances; and (2) on average, the proposed IMOICA obtained an RPI value of 0.041, which is substantially better than the other compared algorithm.

According to the comparison of HV values shown in Table 10, it can be seen that the proposed IMOICA obtained 18 better HV values, which further verify the superiority of the proposed algorithm. Figure 16 shows that the proposed algorithm performs significantly better than the other compared algorithms.

The complexity of IMOICA algorithm is $O(p'q + 2\,mn(s + 1) + m\ \ (f + r) + logv)$. The complexity of the PGDHS algorithm is $O(4\,mn + mnsk + mnsq + 7mns)$, k represents the number of machines. The complexity of HMOEA algorithm is $O(6\,mn(L_1 + L_2 + 1) + 2msk + mnpk)$, L_1 and L_2 is the number of randomly selected individuals. The complexity of PDABC algorithm is $O(m + 7\,mn\ (1 + 4p + s) + mns(L_1 + L_2))$. The complexity of NSGA_II algorithm is $O(5\,mn(1 + p) + 2\,ms(L_1 + L_2 + k))$. By comparison, it is found that the complexity of the proposed IMOICA is smaller. Therefore, for the problem studied, our proposed algorithm is better.

Table 5 Comparison of IMOICA and the exact CPLEX solver

Instance	Surgical case	IGD		HV	
		IMOICA	CPLEX	IMOICA	CPLEX
Inst1	2	**0.011**	0.078	**0.889**	0.510
Inst2	3	**0.033**	0.176	**0.767**	0.461
Inst3	4	**0.024**	0.224	**0.979**	0.678
Inst4	5	**0.126**	0.246	**0.948**	0.777
Inst5	6	**0.044**	0.597	**0.976**	0.624
Inst6	7	**0.037**	0.455	**0.945**	0.742
Inst7	8	**0.003**	0.831	**0.997**	0.810
Inst8	9	**0.096**	–	**0.923**	–
Inst9	10	**0.016**	–	**0.982**	–

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances

5.7 Practical instances

To further verify the effectiveness of the algorithm, we used the experimental data from reference [63] and compared the results with other algorithms, including the

Table 6 Comparison of IGD and HV values obtained by IMOICA and IMOICA_NAR

Scale	IGD		HV		Scale	IGD		HV	
	IMOICA	IMOICA_NAR	IMOICA	IMOICA_NAR		IMOICA	IMOICA_NAR	IMOICA	IMOICA_NAR
4–3	**5.876**	17.34	**0.094**	0.078	10–10	**7.336**	36.04	**0.026**	0.009
4–5	**0.987**	8.153	**0.045**	0.029	30–3	**0.067**	27.84	**0.052**	0.024
4–8	**0.004**	19.43	**0.043**	0.023	30–5	**0.000**	15.56	**0.065**	0.038
4–10	**0.745**	8.574	0.039	**0.064**	30–8	**0.000**	38.37	**0.074**	0.039
7–3	**6.437**	27.34	**0.032**	0.021	30–10	**8.376**	19.82	**0.032**	0.019
7–5	**0.002**	11.58	**0.057**	0.037	50–3	**5.345**	6.325	**0.072**	0.032
7–8	**0.031**	32.42	0.042	**0.063**	50–5	**0.000**	67.32	**0.068**	0.025
7–10	**0.000**	25.80	**0.016**	0.007	50–8	**0.003**	16.07	**0.032**	0.021
10–3	**1.379**	16.72	**0.059**	0.016	50–10	21.03	**19.25**	**0.045**	0.015
10–5	**0.003**	12.22	**0.043**	0.017	50–30	**0.000**	26.62	**0.073**	0.043
10–8	**2.435**	17.73	**0.046**	0.019	–	**–**	–	–	–

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances

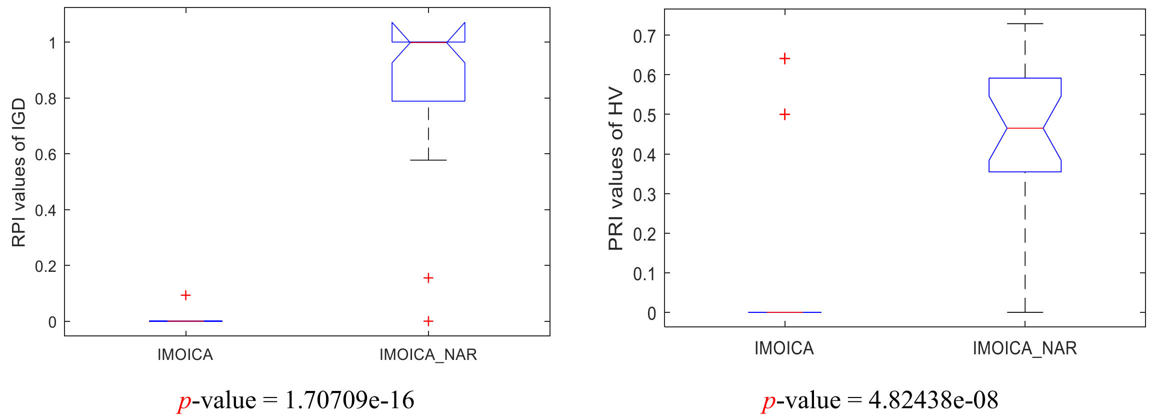


Fig. 12 ANOVA comparison results

Table 7 Comparison of IGD and HV values obtained by IMOICA and IMOICA_NRS

Scale	IGD		HV		Scale	IGD		HV	
	IMOICA	IMOICA_NRS	IMOICA	IMOICA_NRS		IMOICA	IMOICA_NRS	IMOICA	IMOICA_NRS
4–3	**5.234**	17.34	**0.146**	0.072	10–10	**8.321**	34.78	**0.040**	0.002
4–5	**0.442**	6.133	**0.051**	0.031	30–3	**3.436**	28.45	**0.056**	0.032
4–8	**0.013**	11.24	**0.067**	0.017	30–5	**1.367**	32.22	**0.076**	0.039
4–10	**0.653**	9.987	**0.079**	0.035	30–8	**7.891**	35.27	**0.079**	0.031
7–3	**6.737**	26.49	0.044	**0.051**	30–10	**0.000**	21.24	**0.036**	0.011
7–5	**0.009**	29.44	**0.091**	0.066	50–3	**8.973**	11.22	**0.077**	0.033
7–8	**0.032**	55.24	**0.046**	0.039	50–5	**6.234**	53.29	**0.086**	0.025
7–10	**0.001**	12.90	**0.031**	0.027	50–8	**3.457**	34.49	0.032	**0.034**
10–3	**0.986**	17.92	0.073	**0.076**	50–10	**8.673**	23.45	**0.049**	0.021
10–5	**0.011**	19.03	**0.052**	0.034	50–30	**4.729**	35.13	**0.076**	0.039
10–8	**3.478**	24.43	**0.047**	0.022	–	–	–	–	–

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances

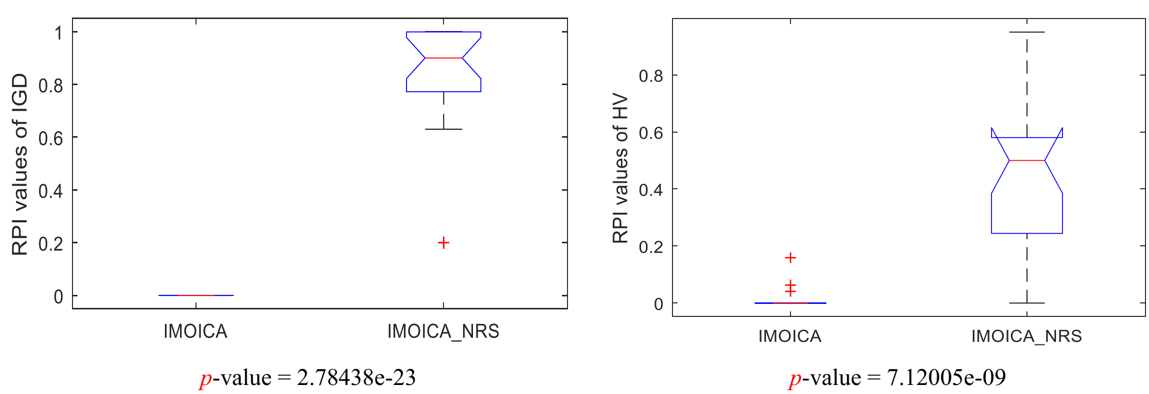


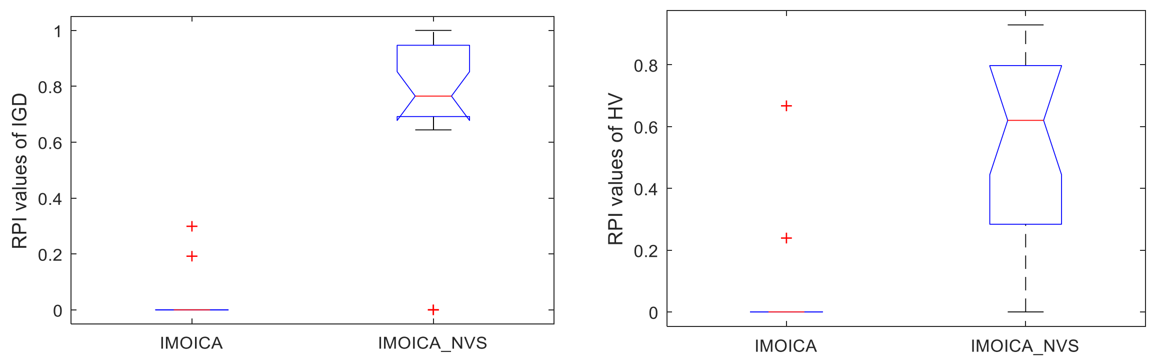
Fig. 13 ANOVA comparison results

PGDHS [48], HMOEA/D [52], P-DABC [73], and NSGA-II [74], to verify the effectiveness of the proposed algorithm. Table 11 shows the number of surgical resources for different types of patients at different stages. The duration time of different types of surgery is shown in Table 12. Table 13 shows the comparative results of the five algorithms based on the above instances. As can be seen from the comparative results reported in Table 13: (1) considering the IGD values, IMOICA obtains 9 better results for 9 instances; (2) for the HV values, IMOICA obtains 8 better

Table 8 Comparison of IGD and HV values between IMOICA and IMOICA_NVS

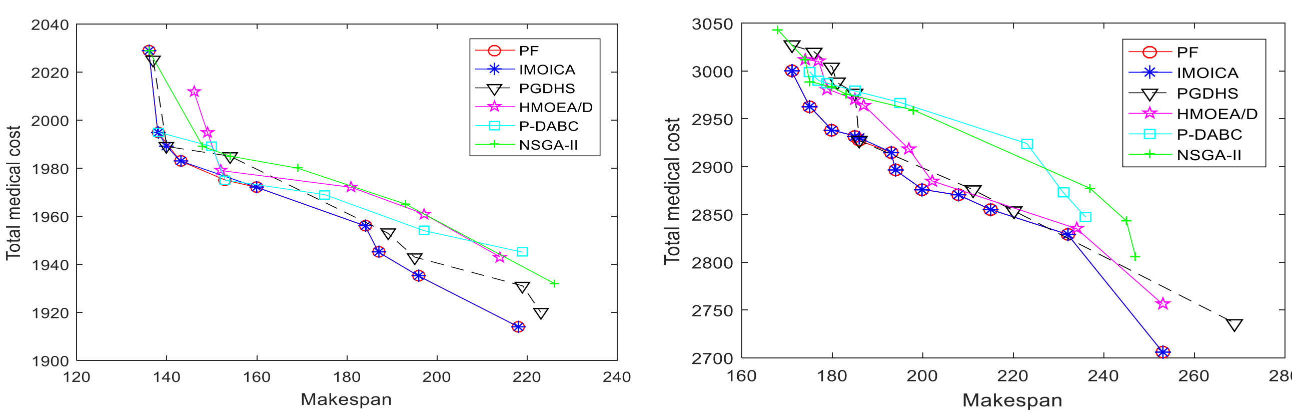
Scale	IGD		HV		Scale	IGD		HV	
	IMOICA	IMOICA_NVS	IMOICA	IMOICA_NVS		IMOICA	IMOICA_NVS	IMOICA	IMOICA_NVS
4–3	**6.324**	21.23	**0.132**	0.080	10–10	**13.25**	56.43	**0.042**	0.029
4–5	**1.456**	8.667	0.046	**0.057**	30–3	**7.891**	22.34	**0.053**	0.032
4–8	**0.679**	12.23	**0.073**	0.043	30–5	**4.374**	24.99	**0.069**	0.038
4–10	**0.774**	17.36	**0.081**	0.050	30–8	**8.563**	34.26	**0.074**	0.041
7–3	**8.342**	23.45	**0.039**	0.021	30–10	**3.719**	22.37	**0.046**	0.033
7–5	**0.002**	31.24	**0.088**	0.046	50–3	**0.000**	3.227	**0.073**	0.039
7–8	**0.367**	62.21	**0.054**	0.028	50–5	**9.876**	36.22	**0.085**	0.055
7–10	**0.012**	8.798	0.027	**0.045**	50–8	6.333	**4.437**	**0.040**	0.034
10–3	0.886	**0.716**	**0.089**	0.067	50–10	**9.432**	31.24	**0.043**	0.038
10–5	**0.045**	13.84	**0.047**	0.039	50–30	**12.37**	37.78	**0.077**	0.043
10–8	**6.341**	22.32	**0.055**	0.042	–	–	–	–	–

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances



p-value = 2.81327e-14 p-value = 8.8721e-08

Fig. 14 ANOVA comparison results IMOICA and IMOICA_NVS



(a) Pareto results for “7-5” (b) Pareto results for “10-5”

Fig. 15 Pareto values of the compared algorithms

results; and (3) the average performance of HV and IGD given from the last line can verify the effectiveness of the IMOICA. Figure 16 represents the Pareto values of the two instances. It should be noted that there is none true Pareto front, because the generated 9 instances are generated by a realistic surgical case scheduling system. Therefore, we

Table 9 Comparison of IGD values between IMOICA, PGDHS, HMOEA/D, P-DABC, and NSGA-II

Scale	Best	IGD					RPI				
		IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II	IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II
4–3	5.667	5.667	17.36	15.63	25.00	25.00	**0.000**	2.064	1.758	3.411	3.411
4–5	1.714	1.714	490.6	122.0	204.7	204.9	**0.000**	285.2	70.17	118.4	118.6
4–8	0.002	0.002	0.230	0.100	0.100	0.005	**0.000**	114.0	49.00	49.00	1.550
4–10	0.825	0.825	860.0	96.91	100.3	151.1	**0.000**	1041	116.4	120.6	182.2
7–3	8.431	8.431	8.431	14.30	10.90	94.20	**0.000**	**0.000**	0.697	0.292	10.17
7–5	0.007	0.007	1.178	0.816	0.866	0.011	**0.000**	180.2	124.6	132.2	0.692
7–8	6.441	9.007	6.441	7.676	8.997	26.44	0.398	**0.000**	0.192	0.397	3.105
7–10	0.019	0.019	18.42	5.812	6.293	3.543	**0.000**	968.7	304.9	330.2	185.5
10–3	2.720	2.720	551.4	648.6	781.1	190.1	**0.000**	201.7	237.4	286.2	68.87
10–5	0.001	0.001	0.329	0.046	0.047	0.058	**0.000**	657.0	91.40	93.00	115.6
10–8	4.035	4.035	780.7	8.657	15.26	20.61	**0.000**	192.5	1.146	2.781	4.107
10–10	9.220	9.220	963.2	824.8	66.17	11.53	**0.000**	103.5	88.46	6.177	0.250
30–3	0.500	0.500	89.19	163.3	72.73	75.43	**0.000**	177.2	325.4	144.3	149.7
30–5	0.551	0.551	221.9	16.04	18.73	10.62	**0.000**	401.8	28.12	33.01	18.27
30–8	2.011	2.011	422.6	18.56	48.96	111.0	**0.000**	209.2	8.227	23.35	54.19
30–10	9.244	9.244	9.972	10.46	10.28	11.00	**0.000**	0.079	0.132	0.112	0.190
50–3	7.033	7.033	7.033	7.074	7.033	20.01	**0.000**	**0.000**	0.006	**0.000**	1.845
50–5	9.087	9.087	971.7	114.1	772. 6	118.5	**0.000**	105.9	11.56	84.02	12.04
50–8	7.168	7.168	951.6	938.9	920.4	975.2	**0.000**	131.8	130.0	127.4	135.1
50–10	23.44	34.02	34.31	23.44	79.88	255.4	0.451	0.464	**0.000**	2.408	9.895
50–30	0.256	0.256	46.77	62.77	55.51	24.00	**0.000**	181.7	244.2	215.8	92.75
Avg	4.684	5.310	307.3	147.6	121.7	110.9	0.041	235.9	87.32	84.43	55.62

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances

used the Pareto solutions obtained by the five compared algorithms as the near Pareto front "PF". It can be seen from the figure that the IMOICA proposed in this paper can get more and better Pareto values. Figure 17 shows the Pareto values of the two different scale practical instances.

5.8 Discussions

The comparison results show that the performance of IMOICA is obviously better than other efficient algorithms for similar scheduling problems. This paper studies the surgical case scheduling problem. In the same running environment, comparing IMOICA with other algorithms, it can be seen that the performance of IMOICA is better.

The principal reasons can be concluded as follow. (1) The concept of AR is introduced into the assimilation strategy, which can enhance the global search ability of the algorithm. Increasing the search ability will help to obtain the global optimal surgical cases scheduling. Through the comparison of the ANOVA diagram of the two algorithms in Sect. 5.3, it can be concluded that IMOICA with the assimilation strategy with AR measure has better performance. (2) The revolution strategy is designed in the proposed algorithm, which can increase the diversity of solutions. By increasing the diversity of the population, more surgical cases scheduling can be generated. Through the comparison of the ANOVA diagram of the two algorithms in Sect. 5.4, it can be concluded that the performance of IMOICA with revolutionary strategy is better. (3) The VNS strategy is embedded in the proposed algorithm, which can keep the IMOICA from falling into a local optimality. In case surgical case scheduling cannot be further optimized. Through the comparison of the ANOVA diagram of the two algorithms in Sect. 5.5, it can be concluded that IMOICA with VNS strategy has better performance. Furthermore, it can be seen from Fig. 2 that the surgical case scheduling is obtained by executing the algorithm, and the scheduling is continuously optimized in the process of optimizing the algorithm.

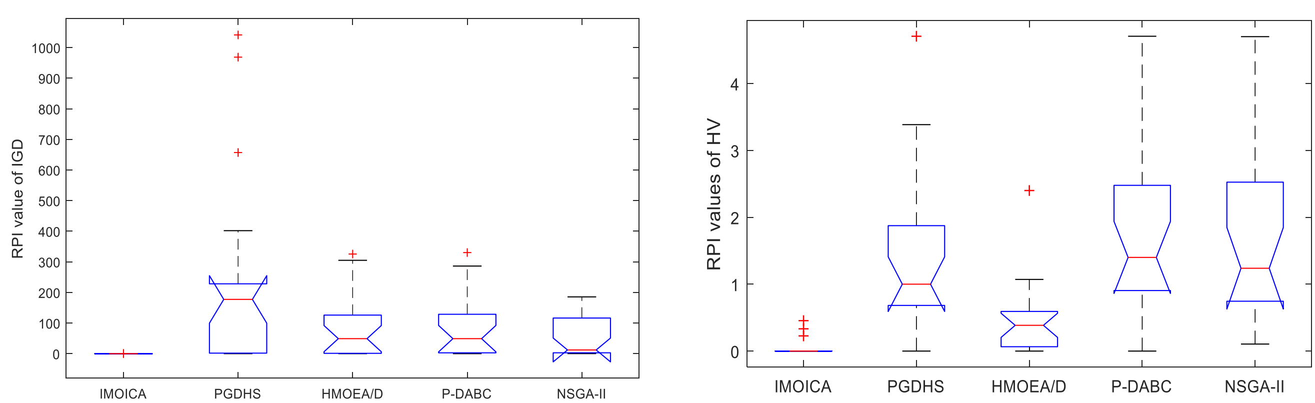
6 Conclusion

Effective surgical case scheduling can significantly improve patient flow, optimize treatment management, and reduce surgical risk. However, the optimization of surgical

Table 10 Comparison of HV values between IMOICA, PGDHS, HMOEA/D, P-DABC, and NSGA-II

Scale	Best	HV					RPI				
		IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II	IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II
4–3	0.097	0.097	0.017	0.091	0.017	0.018	**0.000**	4.706	0.066	4.706	4.389
4–5	0.052	0.052	0.016	0.031	0.010	0.011	**0.000**	2.250	0.677	4.200	3.727
4–8	0.042	0.042	0.015	0.027	0.010	0.022	**0.000**	1.800	0.556	3.200	0.909
4–10	0.038	0.038	0.017	0.024	0.018	0.021	**0.000**	1.235	0.583	1.111	0.810
7–3	0.029	0.029	0.016	0.014	0.010	0.012	**0.000**	0.813	1.071	1.900	1.417
7–5	0.047	0.047	0.045	0.029	0.019	0.021	**0.000**	0.044	0.621	1.474	1.238
7–8	0.032	0.032	0.017	0.022	0.019	0.018	**0.000**	0.882	0.455	0.684	0.778
7–10	0.048	0.036	0.016	0.048	0.020	0.015	0.333	2.000	**0.000**	1.400	2.200
10–3	0.057	0.057	0.013	0.051	0.010	0.010	**0.000**	3.385	0.118	4.700	4.700
10–5	0.036	0.036	0.021	0.026	0.011	0.020	**0.000**	0.714	0.385	2.273	0.800
10–8	0.034	0.034	0.012	0.032	0.022	0.015	**0.000**	1.833	0.063	0.545	1.267
10–10	0.021	0.021	0.014	0.019	0.021	0.019	**0.000**	0.500	0.105	**0.000**	0.105
30–3	0.032	0.022	0.026	0.032	0.011	0.011	0.455	0.231	**0.000**	1.909	1.909
30–5	0.034	0.034	0.017	0.010	0.017	0.012	**0.000**	1.000	2.400	1.000	1.833
30–8	0.054	0.054	0.014	0.031	0.017	0.012	**0.000**	2.857	0.742	2.176	3.500
30–10	0.027	0.022	0.017	0.027	0.014	0.024	0.227	0.588	**0.000**	0.929	0.125
50–3	0.022	0.022	0.011	0.022	0.011	0.011	**0.000**	1.000	**0.000**	1.000	1.000
50–5	0.028	0.028	0.016	0.020	0.012	0.017	**0.000**	0.750	0.400	1.333	0.647
50–8	0.022	0.022	0.022	0.016	0.012	0.017	**0.000**	0.000	0.375	0.833	0.294
50–10	0.045	0.045	0.018	0.037	0.011	0.010	**0.000**	1.500	0.216	3.091	3.500
50–30	0.019	0.019	0.010	0.013	0.016	0.012	**0.000**	0.900	0.462	0.188	0.583
Avg	0.039	0.038	0.018	0.030	0.015	0.016	0.048	1.380	0.443	1.841	1.701

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances



p-value = 5.43997e-05

p-value = 7.46711e-08

Fig. 16 ANOVA comparison results of the five compared algorithms

case scheduling is challenging. This study regards the surgical case scheduling problem as a flexible job shop scheduling problem (FJSP). To tackle the problem, IMOICA was proposed. First, the social hierarchy strategy was adopted to initialize the empire. Then, to enhance the global search ability of the algorithm, the concept of AR was introduced into the assimilation strategy. Furthermore, the revolution strategy was conducted to increase the diversity of population. Finally, a VNS strategy was capable of escaping from being trapped into a local optimum. The proposed algorithm was applied to the surgical case scheduling problem, and the scheduling was made in advance, which shortens the time consumed in the whole surgery process and saves the medical cost.

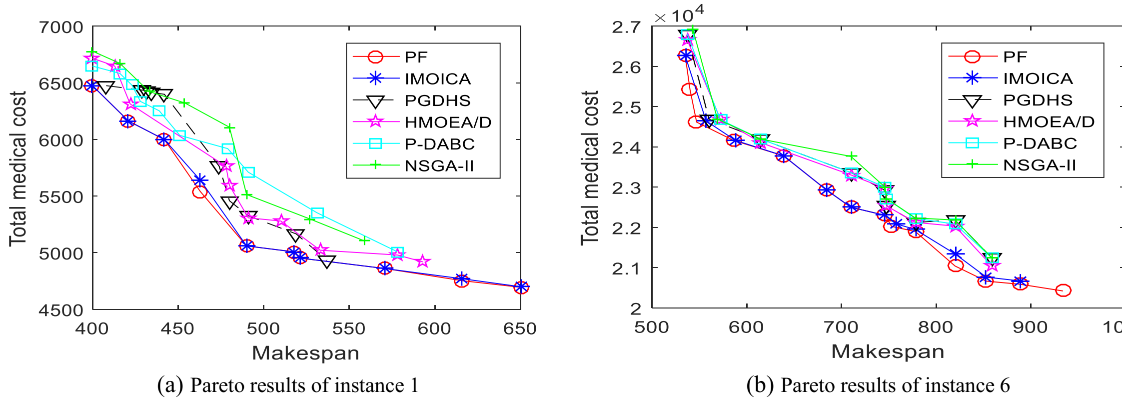


Fig. 17 Pareto values of the different algorithms

Table 11 Different types of instances

Instances	Patients	Surgery type (S: M: L: EL:S)	Pre-operative medicals	Peri-operative medicals	Post-operative medicals
1	8	(2: 4:1:1:0)	5	5	2
2	10	(2: 6:1:1:0)	8	6	4
3	10	(2: 5:2:1:0)	8	6	4
4	15	(3: 9:2:1:0)	10	6	3
5	20	(4: 12:3:1:0)	15	10	4
6	20	(4: 10:3:3:0)	15	10	4
7	30	(7: 16:3:2:2)	19	10	5
8	30	(5: 15:3:4:3)	22	12	5
9	30	(3: 15:3:4:5)	22	12	6

Table 12 Duration of different surgery types

	Pre-operative	Peri-operative					Post-operative
		Small	Medium	Large	E-large	Special	
Duration	Random	Random	Random	Random	Random	Random	Random
	Normal	Normal	Normal	Normal	Normal	Normal	Normal
(min)	(8,2)	(33,15)	(86,17)	(153,17)	(213,17)	(316,62)	(28,17)

The real advantage of IMOICA is that it is a multi-objective optimization algorithm designed according to the research problem, which can solve the research problem with good comprehensive performance, including local search ability, global search ability, convergence and so on. To verify the effectiveness of the proposed method in the research problem, the numerical analysis and experiments was performed. Two performance indicators HV and IGD was introduced to reflect the improvement of the proposed algorithm compared with other algorithms in saving time and cost. First, the proposed algorithm was compared with the CPLEX algorithm, and the effectiveness of the algorithm was verified by two performance indicators. Then, in the strategy verification section, the degree of improvement of the algorithm was further illustrated by the performance indicators and ANOVA diagram. Finally, in the multi-algorithm comparison section, the IMOICA was compared with other classical algorithms to verify its effectiveness.

The problems in the study include rescheduling when emergency patient arrivals were not considered and the uncertain duration of each stage. Therefore, our future research work should start from the following aspects.

Table 13 Duration of different surgical types and stages

Instances	IGD					HV				
	IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II	IMOICA	PGDHS	HMOEA/D	P-DABC	NSGA-II
1	**0.007**	5.341	3.301	8.991	13.46	**0.103**	0.065	0.093	0.054	0.045
2	**2.312**	15.23	13.29	29.34	55.70	**0.096**	0.062	0.083	0.060	0.041
3	**6.332**	27.28	9.211	30.12	18.04	**0.080**	0.072	0.087	0.070	0.062
4	**4.232**	8.223	4.537	4.272	6.291	**0.092**	0.063	0.078	0.082	0.071
5	**1.398**	4.667	4.622	7.352	8.325	**0.088**	0.044	0.053	0.032	0.030
6	**0.233**	9.223	4.237	6.329	2.334	0.036	0.020	0.031	0.003	**0.040**
7	**0.334**	16.32	10.02	17.22	12.30	**0.074**	0.047	0.037	0.033	0.030
8	**2.533**	21.03	29.23	34.09	7.238	**0.163**	0.131	0.042	0.039	0.034
9	**3.212**	34.20	22.33	23.43	33.22	**0.081**	0.062	0.048	0.029	0.073
Avg	2.288	15.72	11.19	17.91	17.43	**0.090**	0.063	0.061	0.045	0.047

The bold in the Table is the optimal values obtained by comparing different algorithms under different instances

First, emergency patient arrival and rescheduling must be considered during the surgery. Second, the time of each stage is represented by fuzzy number. Third, to verify the effectiveness of the solution, applying the solution to other areas such as complex manufacturing automation factories, and considering situations where job scheduling may be affected by many confusing parameters. Moreover, there are three types of scheduling models, we will make an in-depth study of different scheduling models/strategies in the future. For example, in the study of RFID-driven discrete manufacturing system dynamic scheduling with multi-layer network index as heuristic information, the overall and overall optimization of scheduling will be taken into account. When studying the establishment of a series of multi-Agent systems based on blockchain and intelligent contracts, decentralized self-organization will be considered. When considering the block chain of the global optimization model as a two-layer intelligent problem of intelligent manufacturing, the hybrid mode will be adopted.

Funding No funding was provided for the completion of this study.

Data availability All data generated or analyzed during this study are included in this published article (and its additional files).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This study does not violate and does not involve moral and ethical statement.

Informed consent All authors were aware of the publication of the paper and agreed to its publication.

References

1. Zhang, Y., Wang, J., Liu, Y.: Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. J. Clean. Prod. **167**, 665–679 (2017)
2. Burdett, R.L., Kozan, E.: An integrated approach for scheduling health care activities in a hospital. Eur. J. Oper. Res. **264**(2), 756–773 (2018)
3. Chen, H., Zhu, X., Guo, H., Zhu, J., Qin, X., Wu, J.: Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment. J Syst. Softw. **99**, 20–35 (2015)
4. Tang, L., Liu, J., Rong, A., Yang, Z.: A review of planning and scheduling systems and methods for integrated steel production. Eur. J. Oper. Res. **133**(1), 1–20 (2001)
5. Zelen, M.: Optimal scheduling of examinations for the early detection of disease. Biometrika **80**(2), 279–293 (1993)
6. Zhu, H., Hou, M., Wang, C., Zhou, M.: An efficient outpatient scheduling approach. IEEE. Trans. Autom. Sci. Eng. **9**(4), 701–709 (2012)
7. Tsai, C.C., Li, S.H.: A two-stage modeling with genetic algorithms for the nurse scheduling problem. Expert Syet. Appl. **36**(5), 9506–9512 (2009)
8. Pham, D.N., Klinkert, A.: Surgical case scheduling as a generalized job shop scheduling problem. Eur. J. Oper. Res. **185**(3), 1011–1025 (2008)
9. Cardoen, B., Demeulemeester, E., Beliën, J.: Operating room planning and scheduling: a literature review. Eur. J. Oper. Res. **201**(3), 921–932 (2010)
10. Min, D., Yih, Y.: Scheduling elective surgery under uncertainty and downstream capacity constraints. Eur. J. Oper. Res. **206**(3), 642–652 (2010)
11. Marques, I., Captivo, M.E.: Different stakeholders' perspectives for a surgical case assignment problem: deterministic and robust approaches. Eur. J. Oper. Res. **261**(1), 260–278 (2017)
12. Buescher, P.A., Larson, L.C., Nelson, M.D., Jr., Lenihan, A.J.: Prenatal WIC participation can reduce low birth weight and newborn medical costs: a cost-benefit analysis of WIC participation in North Carolina. J. Am. Diet. Assoc. **93**(2), 163–166 (1993)
13. Jebali, A., Alouane, A.B.H., Ladet, P.: Operating rooms scheduling. Int. J. Prod. Econ. **99**(1–2), 52–62 (2006)
14. Al-Refaie, A., Judeh, M., Chen, T.: Optimal multiple-period scheduling and sequencing of operating room and intensive care unit. Oper. Res. **18**(3), 645–670 (2018)

15. Coban, E.: The effect of multiple operating room scheduling on the sterilization schedule of reusable medical devices. Comput. Ind. Eng. **147**, 106618 (2020)
16. Saadouli, H., Jerbi, B., Dammak, A., Masmoudi, L., Bouaziz, A.: A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. Comput. Ind. Eng. **80**, 72–79 (2015)
17. Pan, X., Geng, N., Xie, X., Wen, J.: Managing appointments with waiting time targets and random walk-ins. Omega **95**, 102062 (2020)
18. Li, J.Q., Du, Y., Gao, K.Z., Duan, P.Y., Gong, D.W., Pan, Q.K., Suganthan, P.N.: A hybrid iterated greedy algorithm for a crane transportation flexible job shop problem. IEEE Trans. Autom. Sci. Eng. (2021). https://doi.org/10.1109/TASE.2021.3062979
19. He, X., Pan, Q.K., Gao, L., Wang, L., Suganthan, P.N.: A greedy cooperative co-evolution ary algorithm with problem-specific knowledge for multi-objective flowshop group scheduling problems. IEEE Trans. Evol. Comput. (2021). https://doi.org/10.1109/TEVC.2021.3115795
20. Li, J.Q., Han, Y.Q., Duan, P.Y., Han, Y.Y., Niu, B., Li, C.D., et al.: Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems. J. Clean. Prod. **250**, 119464 (2020)
21. Li, J., Chen, X.L., Duan, P., Mou, J.H.: KMOEA: a knowledge-based multi-objective algorithm for distributed hybrid flow shop in a prefabricated system. IEEE Trans. Ind. Inform. (2021). https://doi.org/10.1109/TII.2021.3128405
22. Gao, K., Cao, Z., Zhang, L., Chen, Z., Han, Y., Pan, Q.: A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. IEEE-CAA J. Automatica Sin. **6**(4), 904–916 (2019)
23. Du, Y., Li, J., Chen, X.L., Duan, P., Pan, Q.: A knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem. IEEE Trans. Emerg. Top. Comput. Intell. (2022). https://doi.org/10.1109/TETCI.2022.3145706
24. Du, Yu., Li J., Li, C., Duan, P.: A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times. IEEE. Trans. Neur. Net. Lear (2022)
25. Li, J., Liu, Z.M., Li, C., Zheng, Z.: Improved artificial immune system algorithm for Type-2 fuzzy flexible job shop scheduling problem. IEEE Trans. Fuzzy Syst. (2020). https://doi.org/10.1109/TFUZZ.2020.3016225
26. Boukef, H., Benrejeb, M., Borne, P.: Flexible job-shop scheduling problems resolution inspired from particle swarm optimization. Stud. Inform. Control. **17**(3), 241–252 (2008)
27. De Giovanni, L., Pezzella, F.: An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. Eur. J. Oper. Res. **200**(2), 395–408 (2010)
28. Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M.: An artificial immune algorithm for the flexible job-shop scheduling problem. Future. Gener. Comp. Sy. **26**(4), 533–541 (2010)
29. Wang, L., Zhou, G., Xu, Y., Wang, S., Liu, M.: An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. Int. J. Adv. Manuf. Tech. **60**(1–4), 303–315 (2012)
30. Rossi, A.: Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. Int. J. Prod. Econ. **153**, 253–267 (2014)
31. Jiang, T., Zhang, C.: Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases. IEEE. Access. **6**, 26231–26240 (2018)
32. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation, pp. 4661–4667. IEEE.
33. Talatahari, S., Azar, B.F., Sheikholeslami, R., Gandomi, A.H.: Imperialist competitive algorithm combined with chaos for global optimization. Commun. Nonlinear Sci. Numer. Simul. **17**(3), 1312–1319 (2012)
34. Karimi, S., Ardalan, Z., Naderi, B., Mohammadi, M.: Scheduling flexible job-shops with transportation times: mathematical models and a hybrid imperialist competitive algorithm. Appl. Math. Model. **41**, 667–682 (2017)
35. Jian, R., Chen, Y., Chen, T.: Multi-parameters unified-optimization for millimeter wave microstrip antenna based on ICACO. IEEE. Access. **7**, 53012–53017 (2019)
36. Zhang, Y., Hu, X., Wu, C.: Improved imperialist competitive algorithms for rebalancing multi-objective two-sided assembly lines with space and resource constraints. Int. J. Prod. Res. **58**(12), 3589–3617 (2020)
37. Mohamed, N., Bilel, N., Alsagri, A.S.: A multi-objective methodology for multi-criteria engineering design. Appl. Soft. Comput. **91**, 106204 (2020)
38. Mahdavi, M., Chehreghani, M.H., Abolhassani, H., Forsati, R.: Novel meta-heuristic algorithms for clustering web documents. Appl. Math. Comput. **201**(1–2), 441–451 (2008)
39. Mahdavi, M., Abolhassani, H.: Harmony K-means algorithm for document clustering. Data. Min. Knowl. Disc. **18**(3), 370–391 (2009)
40. García, M.L.L., García-Ródenas, R., Gómez, A.G.: Hybrid metaheuristic optimization algorithms for time-domain-constrained data clustering. Appl. Soft. Comput. **23**, 319–332 (2014)
41. Jahwar, A.F., Abdulazeez, A.M.: Meta-heuristic algorithms for k-means clustering: a review. PalArch's J. Archaeol. Egypt/Egyptol. **17**(7), 12002–12020 (2020)
42. Abualigah, L., Gandomi, A.H., Elaziz, M.A., Hamad, H.A., Omari, M., Alshinwan, M., Khasawneh, A.M.: Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics **10**(2), 101 (2021)
43. Irfan, M., Jiangbin, Z., Iqbal, M., Arif, M.H.: Enhancing learning classifier systems through convolutional autoencoder to classify underwater images. Soft. Comput. **25**(15), 10423–10440 (2021)
44. Irfan, M., Jiangbin, Z., Iqbal, M., Masood, Z., Arif, M.H., ul Hassan, S.R.: Brain inspired lifelong learning model based on neural based learning classifier system for underwater data classification. Expert. Syst. Appl. **186**, 115798 (2021)
45. Lei, D.: A Pareto archive particle swarm optimization for multi-objective job shop scheduling. Comput. Ind. Eng. **54**(4), 960–971 (2008)
46. Li, J.Q., Pan, Q.K., Chen, J.: A hybrid Pareto-based local search algorithm for multi-objective flexible job shop scheduling problems. Int. J. Prod. Res. **50**(4), 1063–1078 (2012)
47. Wang, L., Wang, S., Liu, M.: A Pareto-based estimation of distribution algorithm for the multi-objective flexible job-shop scheduling problem. Int. J. Prod. Res. **51**(12), 3574–3592 (2013)
48. Gao, K.Z., Suganthan, P.N., Pan, Q.K., Chua, T.J., Cai, T.X., Chong, C.S.: Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling. Inform. Sci. **289**, 76–90 (2014)
49. Wu, X., Sun, Y.: A green scheduling algorithm for flexible job shop with energy-saving measures. J Clean. Prod. **172**, 3249–3264 (2018)
50. Luo, S., Zhang, L., Fan, Y.: Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization. J. Clean. Prod. **234**, 1365–1384 (2019)
51. Li, Z.C., Qian, B., Hu, R., Chang, L.L., Yang, J.B.: An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups. Knowl.-Based Syst. **173**, 83–112 (2019)
52. Wang, L.: Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity

prices. Knowl.-Based Syst. (2020). https://doi.org/10.1016/j.knosys.2020.106177
53. Wallace, B.C., Paul, M.J., Sarkar, U., Trikalinos, T.A., Dredze, M.: A large-scale quantitative analysis of latent factors and sentiment in online doctor reviews. J. Am. Med. Inform. Assoc. **21**(6), 1098–1103 (2014)
54. Afyouni, S., Fetit, A. E., Arvanitis, T. N.: # DigitalHealth: exploring users' perspectives through social media analysis. In: Enabling Health Informatics Applications, pp. 243–246 (2015). https://doi.org/10.3233/978-1-61499-538-8-243
55. Du, S., Hu, Y., Bai, Y., Hu, L., Dong, J., Jin, S., Zhang, H.: Emotional distress correlates among patients with chronic nonspecific low back pain: a hierarchical linear regression analysis. Pain. Pract. **19**(5), 510–521 (2019)
56. Park, S., Woo, J.: Gender classification using sentiment analysis and deep learning in a health web forum. Appl. Sci. **9**(6), 1249 (2019)
57. Jiménez-Zafra, S.M., Martín-Valdivia, M.T., Molina-González, M.D., Ureña-López, L.A.: How do we talk about doctors and drugs? Sentiment analysis in forums expressing opinions for medical domain. Artif. Intell. Med. **93**, 50–57 (2019)
58. Cardoen, B., Demeulemeester, E., Beliën, J.: Optimizing a multiple objective surgical case sequencing problem. Int. J. Prod. Econ. **119**(2), 354–366 (2009)
59. Vijayakumar, B., Parikh, P.J., Scott, R., Barnes, A., Gallimore, J.: A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. Eur. J. Oper. Res. **224**(3), 583–591 (2013)
60. Lee, S., Yih, Y.: Reducing patient-flow delays in surgical suites through determining start-times of surgical cases. Eur. J. Oper. Res. **238**(2), 620–629 (2014)
61. Cappanera, P., Visintin, F., Banditori, C.: Comparing resource balancing criteria in master surgical scheduling: a combined optimisation-simulation approach. Int. J. Prod. Econ. **158**, 179–196 (2014)
62. Al Hasan, H., Guéret, C., Lemoine, D., Rivreau, D.: Surgical case scheduling with sterilising activity constraints. Int. J. Prod. Res. **57**(10), 2984–3002 (2019)
63. Behmanesh, R., Zandieh, M.: Surgical case scheduling problem with fuzzy surgery time: an advanced bi-objective ant system approach. Knowl-Based. Syst. **186**, 104913 (2019)
64. Wang, M., Pan, G.: A novel imperialist competitive algorithm with multi-elite individuals guidance for multi-object unrelated parallel machine scheduling problem. IEEE Access. **7**, 121223–121235 (2019)
65. Lei, D., Yuan, Y., Cai, J., Bai, D.: An imperialist competitive algorithm with memory for distributed unrelated parallel machines scheduling. Int. J. Prod. Res. **58**(2), 597–614 (2020)
66. Sharifi, M., Mojallali, H.: Multi-objective modified imperialist competitive algorithm for brushless DC motor optimization. IETE. J. Res. **65**(1), 96–103 (2019)
67. Zhang, G., Gao, L., Shi, Y.: An effective genetic algorithm for the flexible job-shop scheduling problem. Expert. Syst. Appl. **38**(4), 3563–3573 (2011)
68. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. **6**(2), 182–197 (2002)
69. Hansen, P., Mladenović, N.: Variable neighborhood search: principles and applications. Eur. J. Oper. Res. **130**(3), 449–467 (2001)
70. Lei, D., Li, M., Wang, L.: A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold. IEEE Trans. Cybern. **49**(3), 1097–1109 (2018)
71. Cai, X., Hu, M., Gong, D., Guo, Y.N., Zhang, Y., Fan, Z., Huang, Y.: A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization. Swarm. Evol. Comput. **49**, 178–193 (2019)
72. Zandieh, M., Khatami, A.R., Rahmati, S.H.A.: Flexible job shop scheduling under condition-based maintenance: improved version of imperialist competitive algorithm. Appl. Soft. Comput. **58**, 449–464 (2017)
73. Li, J.Q., Pan, Q.K., Gao, K.Z.: Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. Int. J. Adv. Manuf. Tech. **55**(9), 1159–1169 (2011)
74. Wu, X., Shen, X., Li, C.: The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. Comput. Ind. Eng. **135**, 1004–1024 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Hui Yu received her B.Sc. degree in 2019 from Shandong Normal University, Shandong, China. Since 2019, she has become a postgraduate student in Shandong Normal University, Shandong, China. Her current research interests include discrete optimization and scheduling.

Jun-qing Li received the Master degree of computer science and technology in 2004 from Shandong Economic University, Shandong, China, and Ph.D. degree in 2016 from Northeastern University, Shenyang, China. Since 2004, he was with School of Computer, Liaocheng University. Since 2017, he has been with School of Information Science and Engineering, Shandong Normal University, where he became a Professor in 2017. His current research interests include intelligent optimization and scheduling. He has authored more than 30 refereed papers.

Xiao-long Chen received his B.Sc. degree in 2019 from Shandong Normal University, Shandong, China. Since 2019, he has become a postgraduate student in Shandong Normal University, Shandong, China. His current research interests include discrete optimization and scheduling.

Wei Niu received her B.Sc. degree in 2020 from Shandong Normal University, Shandong, China. Since 2020, she has become a postgraduate student in Shandong Normal University, Shandong, China. Her current research interests include discrete optimization and scheduling.

Hong-yan Sang received her Ph.D. degree in 2013 from Huazhong University of Science and Technology, Wuhan, China. Since 2013, she was with School of Computer, Liaocheng University where appointed as an associate professor in 2016. Her current research interests include discrete optimization and scheduling. She has authored more than 40 refereed papers.