



Research article

Solving the planning and scheduling problem simultaneously in a hospital with a bi-layer discrete particle swarm optimization

Xiuli Wu^{1,*}, Xianli Shen¹ and Linjuan Zhang²

¹ Department of logistics engineering, School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, China

² Department of Anesthesiology, the First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, China

* **Correspondence:** Email: wuxiuli@ustb.edu.cn; Tel: +18801114855; Fax: +18801114855.

Abstract: The operating room is one of the most capital-intensive resources for a hospital. To achieve further improvements and to restrict cost increases, hospitals may need to operate more efficiently with the resources they already possess. The paper considers the joint problem of planning and scheduling patients in operating rooms on an operational level (weekly basis) with two objectives: maximizing the overall patients' satisfaction and minimizing the cost of overtime in operating rooms as well as the daily cost of operating rooms and recovery beds, which is NP-hard. The decision problem is solved using a bi-layer discrete particle swarm optimization, introducing a repair mechanism for infeasible solutions, specific operators like crossover, insertion and exchange. Moreover, a gap finding scheduling heuristic is designed to solve the surgical case sequencing problem. We first compare the performance of the proposed solution method to that of Fei et al. for three instances separately, using data of a Chinese hospital. Next, the efficient Pareto solutions for the joint problem are presented. The results show that the bi-layer discrete particle swarm optimization can solve the operating room scheduling efficiently and effectively.

Keywords: operating room scheduling; integrated planning and scheduling problem; bi-layer discrete particle swarm optimization; patients' satisfaction; operating room cost

1. Introduction

The operating room is one of the most capital-intensive resources for a hospital. Higher quality medical service may lead to higher cost. Hence, the hospital managers need to balance the cost control and the service quality. As one of the core departments, the operating room should be arranged reasonably and the operating room scheduling is the most direct and effective way to balance the cost and the service. For a long time, the head nurse is in charge of scheduling operating rooms in China. The scheduling solution depends on her/his experience. This inefficient and subjective working ways leads to a low utilization of medical resources and times and sometimes even delays a patient's best treatment chance. Therefore, how to schedule the operating room efficiently and effectively is of great importance.

Many studies on the operating room scheduling problem (ORSP) have been carried out. The ORSP has some variants which are featured by the following (1) the type of patients, (2) two levels of ORSP, and (3) the three stages of a surgery. Next, we will review the related studies on these variants.

(1) Studies on the type of patients. According to the severity of the illness, patients are divided into elective patients and emergency patients. The existing related studies focused on one or both type of patient. For example, Chen et al. [1], Wang et al. [2], Bruni et al. [3] focused on the emergency patients. Zhao and Li [4], Neyshabouri and Berg [5], Min and Yih [6] focused on the elective patients. Jebali and Diabat [7], Lamiri [8] focused on both the emergency patients and the elective patients.

(2) The term "ORSP" is used to describe the Surgical Case Assignment Problem (SCAP) and the Surgical Case Sequencing Problem (SCSP). The SCAP is that the surgery date and operating room assigned to each patient selected to be operated on, i.e. the operating room planning problem. Then, the SCSP is that each patient is scheduled for a specific period in the day and operating room, i.e. the operating room scheduling problem. There are some works focusing on only the SCAP. For example, Lamiri [8] developed a stochastic model of the operating room planning problem; they used Monte Carlo simulation and mixed integer programming to solve their model. The goal of their model is to reduce the cost of operating rooms over the long time horizon. Abedini et al. [9] determined a set of elective surgeries over the planning horizon in order to allocate resources. They proposed a multi-step approach and a priority-type-duration (PTD) rule to generate the initial sequence for bin packing to solve this problem. Dellaert and Jeunet [10] explored a variable neighborhood search to solve the planning problem with multiple resources for cardiothoracic surgeries. Similarly, there are many works focusing on the SCSP. For example, Yang et al. [11] focused on the scheduling problem and designed a scheduling method considering surgeons' preferences to the time segments. Time segments and surgeons were seen as two sides of a matching problem. The preference functions of two sides were defined, and the model was built based on the two-sided matching theory. Range et al. [12] addressed the patient admission scheduling problem and proposed an optimization-based heuristic by integrating the branch-and-bound, the column generation, and the dynamic constraint aggregation. Xiang et al. [13] proposed an ant colony optimization approach to efficiently solve operating room scheduling problem based on the knowledge gained in flexible job-shop scheduling problem (FJSP) by observing similarities between the operating room scheduling and a multi-resource constraint FJSP in manufacturing. Riise et al. [14] considered the SCSP, proposed a new generalized model for the problem, showed how this model extended the multi-project, multi-mode resource constrained project scheduling problem with general

time constraints. A search method for solving the proposed model was presented and the algorithm used on-line learning to balance computational loads between a construction and an improvement method, both working on high level solution representations. Duma and Aringhieri [15] proposed an online optimization approach for the Real Time Management (RTM) of operating rooms and evaluated the impact of RTM on the performance of a generic surgical clinical pathway for elective patients.

In recent years, more and more studies on integrated planning and scheduling problem have been appearing. For example, Guerriero and Guido [16] provided a literature review on how operational research can be applied to the surgical planning and scheduling processes. Demeulemeester et al. [17] reviewed the literature on the application of operations research to operating room planning and scheduling. Hallah and Al-Roomi [18] considered the stochastic off-line planning and on-line scheduling of operating rooms, solved the planning and scheduling of operating room with simulation. Aringhieri et al. [19] studied the joint operating room planning and scheduling problem. A 0–1 linear programming formulations were proposed. A two level metaheuristic was developed for solving the problem. Guido and Conforti [20] proposed a multi-objective integer linear programming model aiming at planning and managing hospital operating rooms efficiently. They proposed a hybrid genetic algorithm (GA) to solve the problem. Fei et al. [21] studied the weekly operating room planning and scheduling problem. They solved the problem in two steps. The planning problem was described as a set-partitioning integer-programming model and was solved with a column-generation-based heuristic. The daily scheduling problem, based on the results obtained in the planning phase, was treated as a two-stage hybrid flow-shop scheduling problem (HFSP) and solved with a hybrid GA. Molina-Pariente et al. [22] addressed an integrated operating room planning and scheduling problem, proposed a mixed integer linear programming (MILP) model to optimize the problem and proposed an iterative constructive algorithm for the problem. Landa et al. [23] addressed the problem consisted of two interrelated sub-problems usually referred to as “advance scheduling” and “allocation scheduling”, proposed an integer linear stochastic formulation and developed a hybrid two-phase optimization algorithm to solve the overall problem.

(3) Three stages of a surgery. A surgery usually includes three stages, preoperative (preparing), intraoperative (performing) and postoperative (recovering). Generally, the resources in the preoperative and the postoperative stages are more adequate than that in the intraoperative stage, which determines the efficiency of a hospital to a large extent. Hence many literatures focused only on the resource scheduling problem in the intraoperative stage. For example, Fong et al. [24] pointed out that the intraoperative period was the period in which the surgeons may have the most influence. They systematically reviewed the published efforts to improve intraoperative efficiency, assessed the outcomes of these efforts, and proposed the standardized reporting of future studies. For those considering two stage or three stages, the problem is modelled as the flow shop problem (FSP) or job shop problem. For example, Pham and Klinkert [25] focused on the preoperative, the intraoperative and the postoperative stages, proposed a new operating room scheduling approach based on the multi-mode blocking job shop problem (MMBJS). They formulated the MMBJS as a MILP problem. The problem was solved by the CPLEX solver (<https://www.ibm.com/analytics/cplex-optimizer>). Zhong et al. [26] described the operating rooms as a parallel machine scheduling problem in which a job is processed by multiple machines simultaneously, adopted the two-stage approach to solve the problem and developed a computerized operating rooms system. Burdett and Kozan [27] focused on the

preoperative, the intraoperative and the postoperative stages, introduced a sophisticated FJSP model, where the patients, beds, and health care activities were respectively treated as jobs, machines, and operations. They developed constructive algorithms and hybrid meta-heuristics to solve the FJSP. Abedini et al. [28] considered the intraoperative and postoperative stages. The problem was treated as two-stage FSP and solved with simulation.

Generally, at operational level, the operating room planning problem and the operating room scheduling problem were studied separately. For example, Fei et al. [21] divided the problem into two stages. At the first stage, a weekly surgery planning problem was solved by assigning a date to each surgery. At the second stage, the surgery schedule on each day was solved. Definitely, the assignment decision of the planning problem directly affects the daily scheduling decision, but not vice versa. The method employed in Fei et al. [21] is a decomposition one and it is possible that a bad assignment of surgery date in the first phase will influence the efficiency of the final operating room scheduling. In this paper, the main focus is to present an integrated approach to plan and schedule operating rooms simultaneously. Solving the ORSP in a given week is to determine: (1) the surgery date for each patient, which is referred to as the SCAP; and (2) the operating room and the time assigned to each patient, which is referred to as the SCSP. The SCAP is modelled as a resource-constrained bin-packing problem, which is NP-complete [21]. The SCSP is modelled as a two-stage HFSP with additional consideration of surgeon's constraints. In the SCSP, patients are regarded as the jobs, operating rooms are regarded as the processing machines of the first stage of the HFSP, and the recovery beds are regarded as the processing machines of the second stage of the HFSP. The two-stage HFSP has been proved to be NP-hard [29]. The constraints of the surgeon resource need to be considered. Hence the SCSP is also NP-hard. The ORSP is also NP-hard, which cannot be solved within a limited time especially when the problem size increases. Hence, the heuristics or the meta-heuristics are employed to solve such combinational optimization problem.

Li et al. [30] proposed an energy-aware multi-objective optimization algorithm for solving the HFSP with consideration of the setup energy consumptions. Li et al. [31] proposed a novel discrete artificial bee colony algorithm for solving the multi-objective FJSP with maintenance activities. Liu et al. [32] proposed a hybrid particle swarm optimization (PSO) with estimation of distribution algorithm to solve permutation flow-shop scheduling problem. Li et al. [33] proposed a hybrid variable neighborhood search algorithm for solving the HFSP. Li and Gao [34] proposed an algorithm which hybridizes the GA and tabu search for solving the FJSP. In these algorithms, PSO has the advantages of memory, few parameters, simple structure and easy implementation. Nowadays, it has been well recognized as an efficient method for intelligent search and optimization [35]. For example, Ghorbani et al. [36] proposed a Quantum-behaved PSO, embedded into a multi-layer perceptron technique, to estimate the evaporation rates over a daily forecast horizon. Hence, in this paper, a bi-layer discrete particle swarm optimization (2DPSO) is used to solve the ORSP.

To the best of our knowledge, this is the first attempt to solving the planning and scheduling problem simultaneously with a 2DPSO. The novelties are as following: (1) a 2DPSO is developed to solve the integrated optimization model for the operation rooms planning and scheduling problem; (2) a discrete particle swarm optimization (DPSO) algorithm is proposed to solve the SCAP or the SCSP. There are information exchange and feedback between the two problems; (3) a gap finding method is designed to solve the SCSP; (4) repair mechanisms for infeasible solutions are designed to ensure the feasibility of solutions.

The rest of this paper is organized as follows. Section 2 formulates the integrated optimization

model; Section 3 develops the 2DPSO algorithm; Section 4 carries out a group of experiments; Section 5 is the conclusions.

2. Formulation

Generally, a patient chooses a surgeon at the consultation stage and the surgeon often carry out the surgery for the patient later. Therefore, we only consider that the surgeon for each surgery is determined in advance and cannot be changed. In the period of illness outbreak, a patient often can't be operated immediately after admission. Therefore, the hospital sets the latest surgery date for the patient. The effect of the operation date on the patient's physical condition is different. Some patients waiting for surgery have high risk of infection and need to be operated upon as soon as possible. Thus, the patients are divided into different priority levels to characterize their condition [6]. The patient is divided into r priority levels, and the more urgent the patient needs to be, the higher the priority level. This level is given by the surgeon assigned to each patient. For example, if the risk of infection worsens, the priority of the critical patient is 3, the priority of the patient with cardiovascular and cerebrovascular disease is 2, and the priority of the general patient is 1.

Given that the surgery schedule is usually determined on Friday before the coming week in most hospitals in China, we focus on the surgeries for the elective patients within one week. There are a set of operating rooms and a waiting list of patients to be operated. The task to solve the ORSP in a given week is to determine the surgery date for each patient and the operating room and the time assigned to each patient.

For a patient, the sooner the surgery is performed, the better. For a hospital, the lower the operating room opening costs, the better. An earlier surgery date leads to a higher patient's satisfaction. Increasing patient's satisfaction will inevitably lead to a result that the operating room is open overtime in the first days of one week and is idle in the next days of one week. Therefore, how to balance the patient's satisfaction and the opening time of the operating rooms is of great importance. In this paper, the surgery date for patients is arranged according to the patient's satisfaction and the overtime cost of operating rooms in the SCAP; the operating room and time for patients are scheduled according to the daily operating cost in the SCSP.

To simplify the ORSP, some assumptions are as follows:

- (1) Emergency cases are not taken into consideration because patients admitted from the emergency department are usually carried out immediately and hence only the surgeries for the elective patients are considered.
- (2) Human resources and all material resources except for the recovery beds and the surgeons are always available whenever needed. No surgeon however can operate on more than one patient at the same time; similarly, no recovery bed can be occupied by more than one patient at the same time.
- (3) All operating rooms open simultaneously, and all recovery beds are available at the beginning.
- (4) Once started, an operation cannot be interrupted until it is finished. Moreover, once transferred to a recovery bed, a patient will stay there until the pre-defined recovery time elapses.
- (5) The preparation time for each operation and the clean-up time before leaving the operating room are included in the operating time.

The notations of the variables are listed in Table 1.

Table 1. Notations.

Notations	Description
T	The planning period, a day is indexed by t
J	The number of patients waiting for operations, a patient is indexed by j
M	The number of operating rooms, an operating room is indexed by m
RM	The number of the recovery beds
Q	The number of the departments, a department is indexed by q
L	The number of the surgeons, a surgeon is indexed by l
K_l^t	The allowed maximum surgery time for the l -th surgeon on the t -th day
R_m^t	The number of the regular opening hours of the m -th operating room on the t -th day. If the m -th operating room is unavailable on the t -th day, $R_m^t = 0$
S_m^t	The allowed maximal number of the overtime hours for the m -th operating room on the t -th day
E	The allowed maximal amount of the surgery on every day
N_t	The number of patients assigned to the t -th day
$ET_m^{(t)}$	The completion time of the operation in the m -th operating room on the t -th day
RET_t	The completion time of the operation in the recovery beds on the t -th day
f_t	The maximal scheduled time for all surgeries on the t -th day
D_t	The total duration of all surgeries on the t -th day
ST_{jtl}	The starting time of the j -th patient operated by the l -th surgeon on the t -th day
ET_{jtl}	The ending time of the j -th patient operated by the l -th surgeon on the t -th day
d_l	The date unavailable for the l -th surgeon
s_j	The latest surgery date for the j -th patient
r_j	The priority of the j -th patient
c_j	The duration of the surgery for the j -th patient
p_{jl}	$p_{jl} = 1$ if the l -th surgeon is in charge of the j -th patient, otherwise $p_{jl} = 0$
x_{jm}	$x_{jm} = 1$ if the m -th operating room is available for the j -th patient, otherwise $x_{jm} = 0$
y_{jt}	$y_{jt} = 1$ if the j -th patient is assigned to the t -th day, otherwise $y_{jt} = 0$
y_j	The date of the surgery for the j -th patient
y_{jtl}	$y_{jtl} = 1$ if the j -th patient is in charge by the l -th surgeon and is assigned to the t -th day, otherwise $y_{jtl} = 0$
z_{jmt}	$z_{jmt} = 1$ if the j -th patient is scheduled to be operated on the t -th day in the m -th operating room, otherwise $z_{jmt} = 0$
β	The cost ratio of a regular working hour to an overtime hour, i.e. the penalty cost of the overtime
α	The cost ratio of operating room working hour to recovery room working hour

In the SCAP, for all patients, it is necessary to design a principle to coordinate the operation order. Given the latest surgery date s_j of the j -th patient, the surgery date of the j -th patient is Eq 1.

$$y_j = \sum_{t=1}^T t y_{jt} \quad (1)$$

Hence, the j -th patient's satisfaction degree u_j can be formulated as Eq 2.

$$u_j = \frac{s_j - y_j + 1}{s_j} \quad (2)$$

Combining the j -th patient's priority and his/her satisfaction, the weighted satisfaction for all the patients can be formulated as Eq 3.

$$F_1 = \sum_{j=1}^J r_j u_j \quad (3)$$

The other objective is to minimize the overtime cost of the operating room, which can be formulated in Eq 4.

$$F_2 = \sum_{t=1}^T \sum_{m=1}^M \max \left\{ \beta * (ET_m^{(t)} - R_m^t), R_m^t - ET_m^{(t)} \right\} \quad (4)$$

In the SCSP, the objective is to minimize the daily operating cost including the cost of both the operating rooms and the recovery beds, which can be formulated in Eq 5.

$$f_t = \alpha * \max \left\{ ET_1^{(t)}, ET_2^{(t)}, \dots, ET_M^{(t)} \right\} + RET_t \quad (5)$$

Hence, the integrated optimization model for the ORSP can be formulated as follows.

$$\max F_1 = \sum_{j=1}^J r_j u_j \quad (6)$$

$$\min F_2 = \sum_{t=1}^T \sum_{m=1}^M \max \left\{ \beta * (ET_m^{(t)} - R_m^t), R_m^t - ET_m^{(t)} \right\} \quad (7)$$

$$\min f_t = \alpha * \max \left\{ ET_1^{(t)}, ET_2^{(t)}, \dots, ET_M^{(t)} \right\} + RET_t \quad (8)$$

$$s. t. \sum_{t=1}^T y_{jt} = 1, \forall j \in J \quad (9)$$

$$\sum_{t=1}^T t y_{jt} \leq s_j, \forall j \in J \quad (10)$$

$$y_j > d_l * p_{jl} \text{ or } y_j < d_l * p_{jl}, \forall j \in J, l \in L \quad (11)$$

$$\sum_{j=1}^J y_{jt} \leq E, \forall t \in T \quad (12)$$

$$\sum_{j=1}^J c_j * y_{jtl} \leq K_l^t, \forall t \in T, l \in L \quad (13)$$

$$\sum_{m=1}^M z_{jmt} = 1, \forall j \in J, t \in T \quad (14)$$

$$\sum_{j=1}^J c_j * z_{jmt} \leq (R_m^t + S_m^t), \forall t \in T, m \in M \quad (15)$$

$$ET_{itl} * y_{itl} \geq ST_{jtl} * y_{jtl} \text{ or } ET_{jtl} * y_{jtl} \geq ST_{itl} * y_{itl}, \forall i, j \in J, t \in T, l \in L \quad (16)$$

$$y_{jt} = \begin{cases} 1, & \text{If patient } j \in J \text{ is assigned to date } t \in T \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

$$y_{jtl} = \begin{cases} 1, & \text{If patient } j \in J \text{ whose surgeon is } l \text{ is assigned to} \\ & \text{date } t \in T \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

$$z_{jmt} = \begin{cases} 1, & \text{If patient } j \in J \text{ whose operating date is } t \text{ is assigned to} \\ & \text{operating room } m \in M \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

Eq 6, Eq 7 and Eq 8 are the objectives, the objective function (6) is to maximize the patients' satisfaction, the objective function (7) is to minimize the overtime cost of the operating rooms, and the objective function (8) is to minimize the daily operating cost. Constraint (9) indicates that a patient can only be assigned to a certain date in the planning period. Constraint (10) indicates that the patient's surgery date can't be later than the latest surgery date. Constraint (11) indicates that the surgery date assigned to the patient can't be the date unavailable to his/her surgeon. Constraint (12) indicates that the amount of surgery must not exceed the allowed maximal amount of surgery in every day. Constraint (13) indicates that the maximum surgery time of the surgeon should not exceed the allowed maximal workload for the surgeon in every workday. Constraint (14) indicates that the patient can only be assigned to one operating room. Constraint (15) indicates that the total operating time of the m -th operating room on the t -th day would not exceed the allowed maximal opening hours of the operating room. Constraint (16) indicates that the surgeon can't carry out two operations at the same time. Constraints (17), (18), (19) are about the decision variables.

3. The bi-layer discrete particle swarm optimization (2DPSO) algorithm

As described in Section 2, the SCAP is to provide a surgery date for each patient and the SCSP is to determine the sequence of operations in each operating room in each day. The SCAP is affected by the feedback obtained from the SCSP, while SCSP is based on the results obtained from the SCAP. The SCAP is modelled as a resource-constrained bin-packing problem and solved by a discrete particle swarm optimization (DPSO4A) algorithm. The SCSP is modelled as a two-stage HFSP with additional consideration of surgeon's constraints and solved by a discrete particle swarm optimization (DPSO4S). The ORSP is NP-hard, which cannot be solved within a limited time especially when the problem size increases. We propose a bi-layer discrete particle swarm optimization algorithm (2DPSO) to solve the integrated optimization model. The structure of the 2DPSO is shown in Figure 1. The result of DPSO4A algorithm is regarded as the input of the DPSO4S. The scheduling result obtained by the DPSO4S algorithm is used as the feedback guiding the assignment of the SCAP.

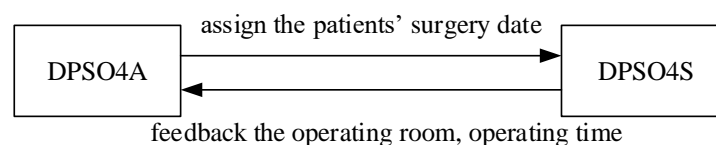


Figure 1. The structure of the 2DPSO.

3.1. The PSO algorithm

Particle Swarm Optimization (PSO) is a swarm intelligence algorithm proposed by Kennedy and Eberhart in 1995 [37]. In this algorithm, each particle of the swarm represents a solution in the solution space, and the particles adjust their flight according to their flight experience and the peer's flight experience. In the course of the evolution, each particle maintains its own historical best position, i.e. the local best; the swarm maintains the historical best position, i.e. the global best. By evaluating the fitness of the particles, the local best and the global best are updated. Each particle updates itself continuously through the local best, the global best and its own flight speed. Thereby a new generation of the swarm is generated. The flowchart of the PSO is shown in Figure 2. The particle's flight speed and its location are updated according to Eq 20 and Eq 21.

$$V_i^{k+1} = w \times V_i^k + c_1 \times rand_1 \times (pB_i^k - x_i) + c_2 \times rand_2 \times (gB^k - x_i) \quad (20)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (21)$$

Where V_i^{k+1} is the i -th particle's flight speed in the k -th iteration, which is limited in the interval $[V_{min}, V_{max}]$; x_i is the i -th particle's location in the k -th iteration; pB_i^k is the local best of the i -th particle; gB^k is the global best; $rand_1, rand_2$ are a random number in $[0, 1]$; w is the inertia weight, which controls the influence of the previous speed on the current speed and balances the ability of the algorithm to explore and exploit in the search process; c_1 is the cognitive coefficient to adjust particles moving to the pB_i^k ; c_2 is the social coefficient to adjust particles moving to the gB^k .

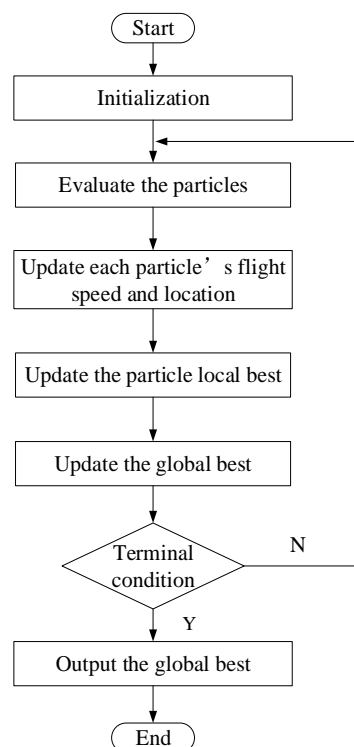


Figure 2. The flowchart of PSO.

The basic PSO algorithm is mainly suitable for solving continuous optimization problems. However, the ORSP is a combinational optimization problem. Therefore, a 2DPSO algorithm is designed to solve the ORSP referring to the discrete particle swarm optimization algorithm (DPSO) by Pan et al. [38]. The flowchart of 2DPSO is shown in Figure 3. The details are described in Section 3.2 and Section 3.3.

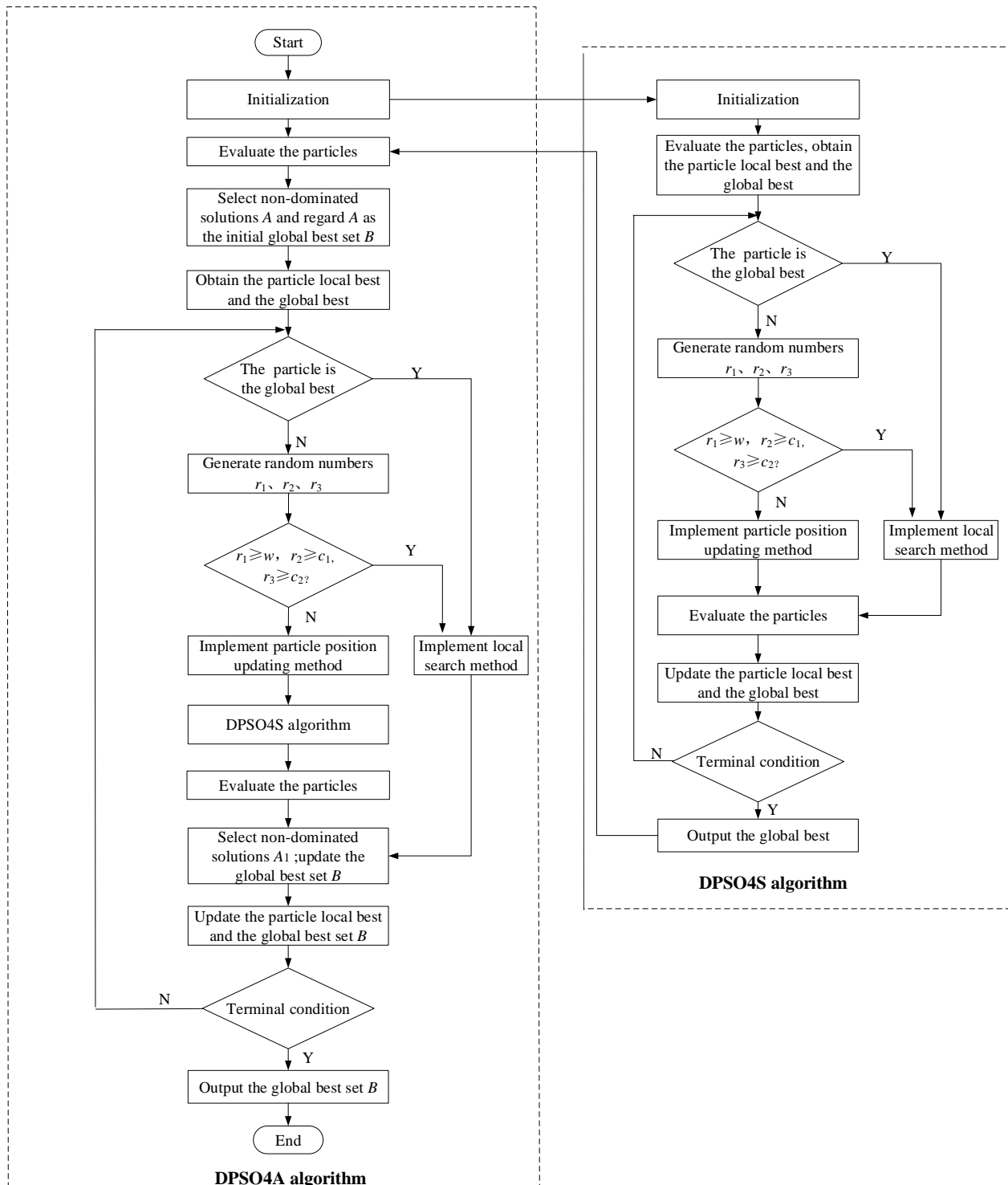


Figure 3. The flowchart of 2DPSO.

3.2. The DPSO4A algorithm

The steps of the PSO4A algorithm are as follows.

Step 1. Initialize parameters (w , c_1 , c_2 , the size of swarm S and the number of cycles P); initialize a swarm, each particle in which represents the date assigned to each patient's surgery. In this process, it is possible to generate an infeasible solution. For example, the work schedule of a patient's surgeon conflicts with the assignment results. Therefore, it is necessary to design a particle legalized function as shown in Section 3.2.2.

Step 2. For each particle, the scheduling results are obtained with the DPSO4S algorithm.

Step 3. Calculate the patients' satisfaction according to Eq 3, the overtime cost of the operating rooms according to Eq 4, which is determined with the DPSO4S.

Step 4. Select all the non-dominated solutions A and regard A as the initial global best set B .

Step 5. Find the particle local best; select a solution as the global best from the global best set B .

Step 6. Updating.

Step 6.1. Update the parameters (w , c_1 , c_2) and the particles' position. Similarly, in this step, the updated particles may be infeasible solutions, it is necessary to correct infeasible solutions with the particle legalized function as shown in Section 3.2.2;

Step 6.2. Evaluate the two objectives of the updated particles;

Step 6.3. Select all the non-dominated solutions A_1 in the updated particle swarm, update the global best set B ;

Step 6.4. Update the particle local best and the global best;

Step 6.5. Increase iteration counter.

Step 7. If the terminal condition is satisfied, stop the iteration and output the global best set B ; otherwise return to the Step 6 to continue a new iteration.

The details of the DPSO4A are designed as follows.

3.2.1. The particle representation

Each particle is a feasible assignment of all patients' surgery date. We represent it as an integer vector of J . Each position corresponds to a patient, and the value in each position is the assigned date for the patient. An example is shown in Figure 4. There are 10 patients waiting for operating, the planning horizon is 5 weekdays, the first patient is assigned to Friday, the second patient is assigned to Thursday, and so on.

1	2	3	4	5	6	7	8	9	10
5	4	4	1	2	2	3	4	1	5

Figure 4. An example of the particle representation.

3.2.2. The particle legalized function

There are three Constraints (10), (11), (13) in the SCAP, both the initial particles and the updated particles may be infeasible. Therefore, it is necessary to design a particle legalized function. The details are as follows.

Step 1. According to Constraints (10), (11) to find out the infeasible assignment for patients of a particle.

Step 2. For an infeasible assignment for the patient i with a surgery date t_0 , reassign surgery date t meeting both Constraints (10), (11), in addition, in order to ensure the balance of the number of surgery on every day, if there is an assignment for the patient j ($j \neq i$) who is operated on date t and $t_0 \notin d_l(p_{jl} = 1)$ and $t_0 \geq s_j$, then select the assignment for the patient j and replace the surgery date with t_0 , otherwise, don't replace, correct the next infeasible assignment. Repeat until all the infeasible assignments are corrected.

Step 3. For the corrected particle, sum the workload for each surgeon.

Step 4. Find out the surgeon l that is overloaded with work and the corresponding date t_1 .

Step 5. A patient is randomly selected from all patients assigned to date t_1 and operated by surgeon l , assign the patient to the date meeting the constraint of both Constraints (10), (11) with the least amount of surgery. Repeat until all surgeons are corrected in every day.

Step 6. Output the feasible solution.

3.2.3. The particle location updating method

In PSO, particles update their locations by flying. Hence, a method is designed to update the particle location.

(1) Updating method

The location updating method is as shown in Eq 22:

$$X_i^{k+1} = c_2 G_3(c_1 G_2(w G_1(X_i^k), pB_i^k), gB^k) \quad (22)$$

The location updating equation is divided into three parts, namely:

(a) Free fly

$$A_i^k = w G_1(X_i^k) = \begin{cases} G_1(X_i^k) & rand < w \\ X_i^k & rand \geq w \end{cases} \quad (23)$$

It indicates that particles will consider their own flight speed. The Insert operator is applied such that the position of the particle is updated to a better position by sharing knowledge with itself.

Hence, $G_1(X_i^k)$ is implemented with “Insert” method.

Insert: Select two positions a , b ($b > a$) randomly, the information at position b is inserted into the front of the information at position a . An example is shown in Figure 5.

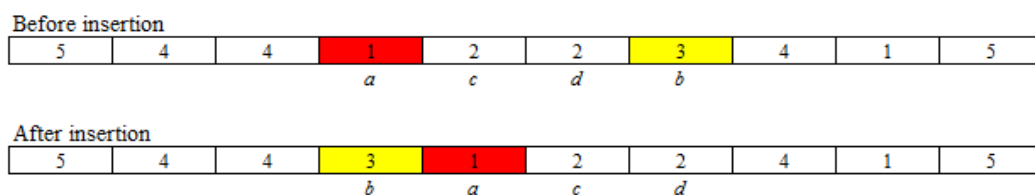


Figure 5. An example for the Insert method.

(b) Learn from the swarm

$$B_i^k = c_1 G_2(A_i^k, pB_i^k) = \begin{cases} G_2(A_i^k, pB_i^k) & \text{rand} < c_1 \\ A_i^k & \text{rand} \geq c_1 \end{cases} \quad (24)$$

It indicates that the particles adjust the location according to the particle local best location. The crossover operator is applied such that the position of the particle is updated to a better position by sharing knowledge with the local best particle. Hence, $G_2(A_i^k, pB_i^k)$ is implemented with “Crossover” method.

(c) Learn from the history

$$X_i^{k+1} = c_2 G_3(B_i^k, gB^k) = \begin{cases} G_3(B_i^k, gB^k) & \text{rand} < c_2 \\ B_i^k & \text{rand} \geq c_2 \end{cases} \quad (25)$$

It indicates that the particles adjust the location according to the global best location. The crossover operator is applied such that the position of the particle is updated to a better position by sharing knowledge with the global best particle. Hence, $G_3(B_i^k, gB^k)$ is implemented with “Crossover” method.

Crossover: A fragment is randomly selected from the particle X to replace the corresponding location of the particle local best (or the global best), and the replaced particle local best (or the global best) is taken as the updated particle X. An example is shown in Figure 6.

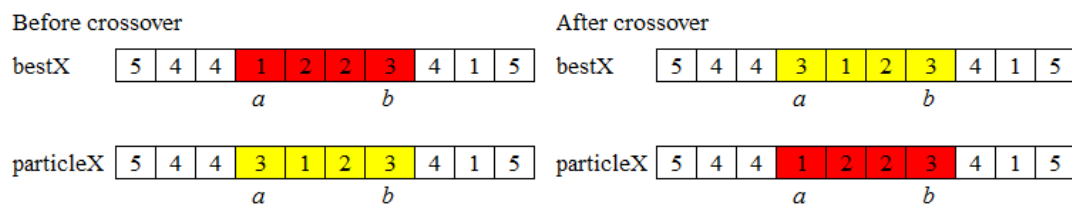


Figure 6. An example for the Crossover method.

(2) The adaptive parameters' setting

w , c_1 and c_2 are adjusted by adaptive adjustment method, the specific operations are as follows.

The inertia weight w is used to adjust the influence of the previous speed on the current speed, which can be set to a certain value ranging from 0 to 1. In the DPSO4A algorithm, to balance the ability of global exploration and local exploitation in the search process, w decreases linearly with the number of iterations increasing. While w is large, the particle retains more historical information, which is beneficial to the diversity of the swarm, thereby the DPSO4A algorithm has the strong ability of global exploration; on the contrary, the particle retains less historical information, which is beneficial to the concentration of the swarm, thereby the DPSO4A algorithm has the strong ability of local exploitation [38]. Eq 26 formulates this setting.

$$w = w_0 - \frac{w_e \times k}{P} \quad (26)$$

Where w_0 is the initial value; w_e is the total length of inertia weight declining; k is the current number of iterations; P is the total number of iterations.

The cognitive coefficient c_1 is to adjust particles moving to the pB_i^k which is usually set to a fixed value of 2 in the basic PSO algorithm [37]. In order to balance the diversity and convergence of the DPSO4A algorithm dynamically, an adaptive method is used to adjust the value of c_1 [38]. The distance d between two particles is expressed by the number of different information of the current particle and the particle local best at the same location. When the distance is large, in order to maintain the concentration of the swarm, we should increase the value of c_1 , accelerate the speed of particles moving to the particle local best; when the distance is small, in order to maintain the diversity of the swarm, we should reduce the value of c_1 , slow down the speed of particles moving to the particle local best.

$$c_1 = \frac{d}{n} c_1^0 \quad (27)$$

Where n is the length of the particle; d is the distance between two particles (the current particle and the particle local best); c_1^0 is an initial value.

The social coefficient c_2 is to adjust particles moving to the gB^k , which is usually set to a fixed value of 2 in the basic PSO algorithm [37]. In order to balance the diversity and convergence of the DPSO4A algorithm dynamically, c_2 is adjusted by an adaptive method similar to the adaptive method of c_1 [38].

$$c_2 = \frac{d}{n} c_2^0 \quad (28)$$

Where n is the length of the particle; d is the distance between two particles (the current particle and the global best); c_2^0 is an initial value.

(3) Local search

In order to improve the local search ability of the algorithm, when the particle i is the global best or the location updating equation doesn't work, a local search operator should be performed. The local search operator is to exploit a better particle with the neighbour of the original particle, which only needs to make some minor changes to the particles. Hence, the local search is implemented with "Exchange" method.

Exchange: The exchange method is to generate randomly two positions (a and b), and exchange their value. An example is shown in Figure 7.

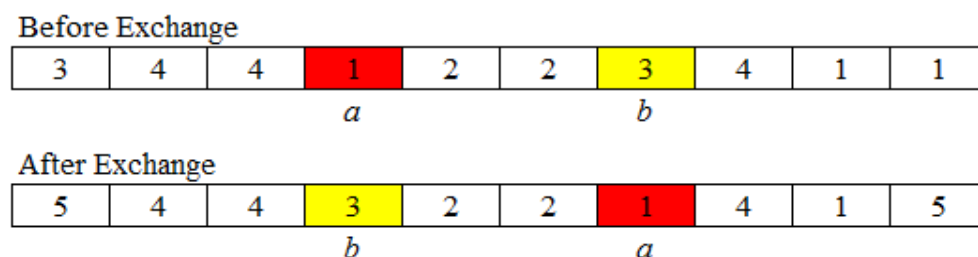


Figure 7. An example of the exchange method.

3.3. The DPSO4S algorithm

The DPSO4S algorithm works in the inner layer of the 2DPSO to solve the SCSP. The SCSP is modelled as a two-stage HFSP with the additional consideration of surgeon's constraints. The steps of the DPSO4S algorithm are as follows.

Step 1. Initialize parameters (w , c_1 , c_2 , S and P); initialize a swarm, each particle of which represents a daily surgery schedule. The procedure for generating the initial swarm is as follows:

Step 1.1. Given that patients are sorted in ascending order of admission time, they will be first successively scheduled into a random-chosen operating room available on the given date;

Step 1.2. When all patients have been scheduled into the operating rooms, they are re-sorted by their completion time in ascending order;

Step 1.3. Each of them is scheduled to a randomly-chosen available recovery bed in the recovery room. It is possible that a patient is scheduled to a recovery bed while he/she comes around in the operating room. The reason is that the recovery bed is not always available. In this case, when the assigned recovery bed is available, he/she is moved from the operating room to the recovery room immediately.

Step 2. Decode the particles with the scheduling heuristic, the details of the scheduling heuristic are shown in Section 3.3.3; calculate the objective; find the particle local best and the global best based on the objective.

Step 3. Updating.

Step 3.1. Update the parameters (w , c_1 , c_2) and the particle's location. In this process, it is possible to generate an infeasible solution. It is necessary to design a particle legalized function as shown in Section 3.3.2;

Step 3.2. Calculate the objective of the updated particles;

Step 3.3. Update the particle local best and the global best;

Step 3.4. Increase iteration counter.

Step 4. If the terminal condition is satisfied, stop the searching and output the global best; otherwise return to the Step 3 to continue a new iteration.

The details of the DPSO4S are as follows.

3.3.1. The particle representation

Each particle, i.e. a feasible daily surgery schedule, is represented as an integer array, consisting of three parts:

- (1) V_1 : A vector of size J_t (the number of patients assigned to the t -th day), storing the number of patients assigned to the t -th day. Each member of V_1 corresponds a patient.
- (2) V_2 : A vector of size J_t , containing the indexes of the operating rooms in the same order as that in V_1 .
- (3) V_3 : A vector of size J_t , containing the indexes of the recovery beds in the same order as the patients given in V_1 .

Figure 8 shows an example of the representation for a scheduling problem. There are 5 patients waiting for operation on the t -th day, the first patient is assigned to the first operating room and the first recovery bed, the second patient is assigned to the second operating room and the first recovery bed. Similarly, the fifth patient is assigned to the third operating room and the third recovery bed.

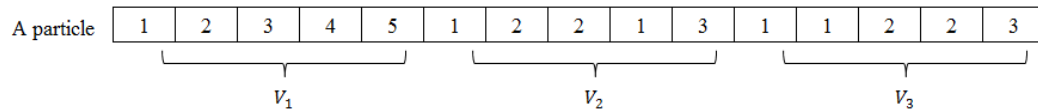


Figure 8. An example of the particle representation for DPSO4S.

3.3.2. The particle legalized function

In the SCSP, in the process of updating, it is possible to generate infeasible solutions. Therefore, it is necessary to design a particle legalized function. The details are as follows.

Step 1. Find out all infeasible assignment for patients who were assigned to unavailable operating rooms.

Step 2. For an infeasible assignment for patient i with operating room m_0 , reassign patient i to the available operating room m , in addition, in order to ensure the balance of the number of surgeries in each operating room, if there is a patient j ($j \neq i$) who is operated on operating room m , then select patient j and replace the operating room with m_0 , otherwise, do not replace, correct the next infeasible assignment for patients. Repeat until all infeasible assignments are corrected.

Step 3. For the corrected particle, sum the workload assigned to each operating room.

Step 4. Find out the operating rooms in which there is overwork.

Step 5. Select a patient randomly from the assigned operating room with overwork and assign him/her to an operating room with the least amount of surgery in all operating rooms available to the patient. Repeat until all the overloading operating rooms are corrected.

Step 6. Output the feasible solution.

3.3.3. The scheduling heuristic

Wu et al. [39] proposed the gap finding method to make a scheduling solution. Before arranging the job's operation, the status of a machine is checked if it has an idle time from the beginning to the current time to insert into the current operation. This method pushes the job's operation forward continuously. As long as the machine is idle, the job's operation is inserted as early as possible, so that the operation on each machine can be arranged in a compact manner to achieve active scheduling. The gap finding method has been proved to be an effective heuristic to minimize the makespan [39]. Hence, we design a gap finding method to solve the SCSP. Before arranging the starting time ST_j of the patient j , the surgeon's available gap and the available gap of operating room assigned to patient j until now needs to be calculated first and then calculate ST_j according to the following four cases. The flowchart of the scheduling heuristic is shown in Figure 9.

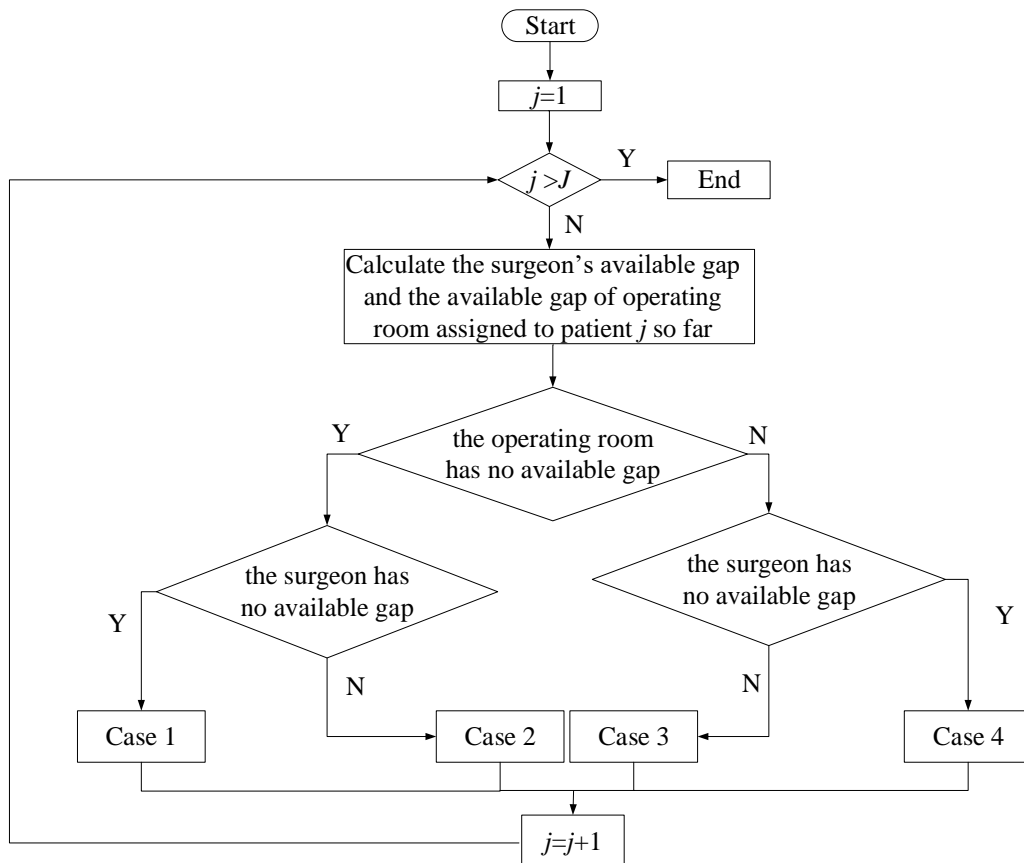


Figure 9. The flowchart of the scheduling heuristic.

Table 2. Variable.

abbreviations	Full name
OR_et	The operation ending time in the operating room
S_et	The operation ending time of the surgeon
OR_est	The earliest starting time of the operating room
S_est	The earliest starting time of the surgeon
U_sg	The upper bound of the surgeon's gap
L_sg	The lower bound of the surgeon's gap
U_org	The upper bound of the operating room's gap
L_org	The lower bound of the operating room's gap

Case 1. If neither the surgeon nor the operating room has available gap.

$$\text{Set } ST_j = \max \{OR_et, S_et\}$$

Case 2. If the surgeon has available gaps, the operating room has no available gap.

Set the operation ending time in the operating room as the earliest starting time for the patient and then search the available gap of the patient after the earliest starting time. If an available gap is

found, insert patient j into the gap, otherwise calculate ST_j according to Case 1. The detailed steps are as follows.

Step 1. Start from the first available gap of the surgeon.

Step 2. Calculate a , $a = \{U_{sg} - \max \{OR_{est}, L_{sg}\}\}$.

If $a > c_j$, insert the patient into the gap, $ST_j = \max \{OR_{est}, L_{sg}\}$, otherwise, compare the next gap until all the gaps are compared. Finally, if all gaps are unavailable, calculate ST_j according to Case 1.

Case 3. If the operating room has available gaps, the surgeon has no available gap.

Set the operation ending time of the surgeon as the earliest starting time for the patient and then search the available gap of the patient after the earliest starting time. If an available gap is found, insert patient j into the gap, otherwise calculate ST_j according to Case 1. The detailed steps are as follows.

Step 1. Start from the first available gap of the operating room.

Step 2. Calculate a , $a = \{U_{org} - \max \{S_{est}, L_{org}\}\}$.

If $a > c_j$, insert the patient into the gap, $ST_j = \max \{S_{est}, L_{org}\}$, otherwise, compare the next gap until all the gaps are compared. Finally, if all gaps are unavailable, calculate ST_j according to Case 1.

Case 4. If both the operating room and the surgeon have available gaps.

We match all the gaps of the surgeon with all the gaps in the operating room. If an available common gap is found, then insert patient j into the common gap, otherwise follow the Case 1. The detailed steps are as follows.

Step 1.

If $\max \{U_{org}\} > S_{et}$, update the surgeon's gaps as follow: Set $[S_{et}, \max \{U_{org}\}]$ as a surgeon's gap;

If $\max \{U_{sg}\} > OR_{et}$, update the operating room gaps as follow: Set $[OR_{et}, \max \{U_{sg}\}]$ as an operating room's gap.

Step 2. Obtain the common gaps of the surgeon's gaps and the operating room gaps.

Step 3. Start from the first common gap.

Step 4. Calculate a , b . $a = \max \{L_{org}, L_{sg}\}$, $b = \max \{U_{org}, U_{sg}\}$.

Common gap = $b - a$.

If $b - a > c_j$, insert the common gap, $ST_j = a$, otherwise, compare the next common gap until all the gaps are compared. Finally, if all common gaps are unavailable, ST_j is calculated according to Case 1.

To explain the gap finding method clearer, an example is shown in Figure 10. There are 5 patients waiting to be operated; the duration of the surgery for these patients is 3, 2, 3, 3 and 3 respectively; the surgeon for these patients is 1, 1, 1, 2 and 2 respectively; the operating room for these patients is 1, 2, 3, 1 and 2. The numbers in each rectangle indicate "patient number" and the same filled color indicates patients operated by same surgeon. Patients 1, 2, 3 and 4 have been sequenced and now it is the patient 5 turn to be sequenced. As can be seen from Figure 10, if we don't consider the existed idle time slots, the patient 5 is placed after the patient 2. However, if the

idle time slot is considered, the patient 5 is placed before the patient 2. The complete time for the patient 5 will be greatly reduced and the utilization of operating rooms will be increased.

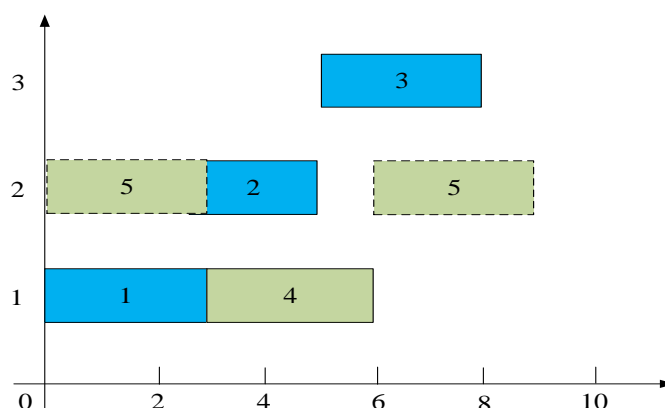


Figure 10. An example of how the gap finding method works.

4. Experiment study

4.1. Design of experiments

4.1.1. Experiment setting

All the experiments are conducted in a desktop computer with an Inter Core i5-3230 2.60 GHz CPU, 4.0RAM, WIN-7 OS, and Matlab©.

4.1.2. The experiment data

In order to evaluate the performance of the proposed approach in improving the practical arrangement of surgical cases in the operating theatre, real data of a hospital in China, are used in this study. Normally, all the operating rooms are open from 8:00 am to 4:00 pm. The recovery room open simultaneously with operating rooms and remain open until all patients transferred out of the operating room are recovered.

In this study, the experiments are based on 16383 data from the operating theatre, which were provided by a hospital in China over a 5-months period (from June 1, 2015 to October 5, 2015). The data mainly consists of date of surgery, the starting time and the ending time of surgery, time of the patient's leaving recovery room, corresponding specialty, surgeon, surgery assistants and nurses for each surgical case together with some personnel information (such as the patient's birth date, gender, etc.). In principle, these all 16383 data can be used. In order to verify the superiority of the proposed method, we select the data in one week with a large number of patients waiting to be operated. In order to evaluate the performance of the proposed method in the round, three experiments with different size (the small-size instance, the medium-size instance and the large-size instance) are carried out. The details of three experiments are listed in Table 3.

Table 3. Details of three experiments.

Instance	Department	Patient	Surgeon	Operating room	Recovery bed
The small-size instance	5	28	12	2	2
The medium-size instance	7	75	21	4	4
The large-size instance	10	111	36	8	8

4.1.3. Parameters setting

The parameters are set as Table 4. The penalty cost of the overtime β is the same as that in Tessler et al. [40] and the cost ratio of operating room working hour to recovery room working hour α is the same as that in Schuster et al. [41]. The inertia weight w is generally set to decrement from 0.9 to 0.4 linearly [42]. Hence, the initial inertia weight w_0 and the total length of inertia weight declining w_e are set to 0.9 and 0.5 respectively. In the 2DPSO algorithm, the cognitive coefficient c_1^0 or the social coefficient c_2^0 controls the probability of performing the “Crossover” method. Hence, c_1^0 and c_2^0 is the same as crossover probability P_c in Wu and Sun [43]. PSO doesn't require a large swarm size. Generally the size of swarm S takes 20–40 to achieve good solutions [42]. Hence, the size of swarm S is set to 30. The number of cycles P is determined by the decision problem. In order to balance the quality of solutions and the computational time, the number of cycles P is set to 100 by experiment.

Table 4. Parameters setting.

parameter	Value
β	1.5 [40]
α	10.9 [41]
w_0	0.9
w_e	0.5
c_1^0	0.8
c_2^0	0.8
S	30
P	100

4.1.4. The aim

In order to evaluate the performance of the proposed method, two groups of experiment are carried out.

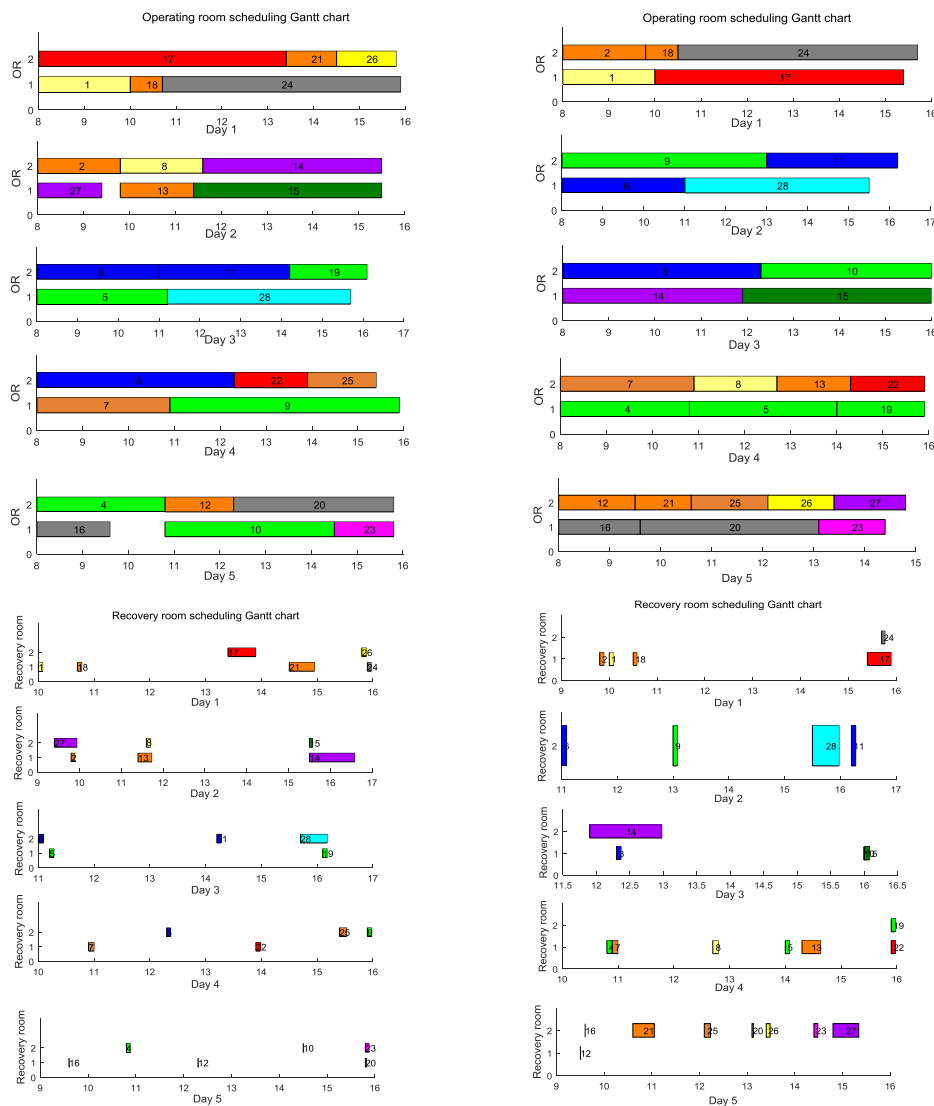
- (1) Comparative experiment. The resources and constraints considered in Fei et al.' study [21] are the same as those considered in our study and Fei et al.'s experiment results [21] are representative. Hence, we recode the algorithm in Fei et al. [21] and compare its performance with the proposed method on solving three instances to prove the superiority of 2DPSO.
- (2) Numerical experiment for practical example. Design experiments with the practical example to solve bi-objective problem, to prove the diversity of solutions.

4.2. Results

4.2.1. Comparative experiment

In the following section, the small-size instance, the medium-size instance, and the large-size instance experiments have been carried out, scheduling Gantt charts are obtained and compared with the scheduling Gantt chart in Fei et al. [21]. Then, the scheduling results are presented in Table 5. Due to the medium-size instance scheduling results table and the large-size instance scheduling results table have too much data, we put these two tables in Appendix A.

(1) The small-size instance numerical experiment result



The solution obtained by the proposed method

The solution obtained by Fei et al.' [21] method

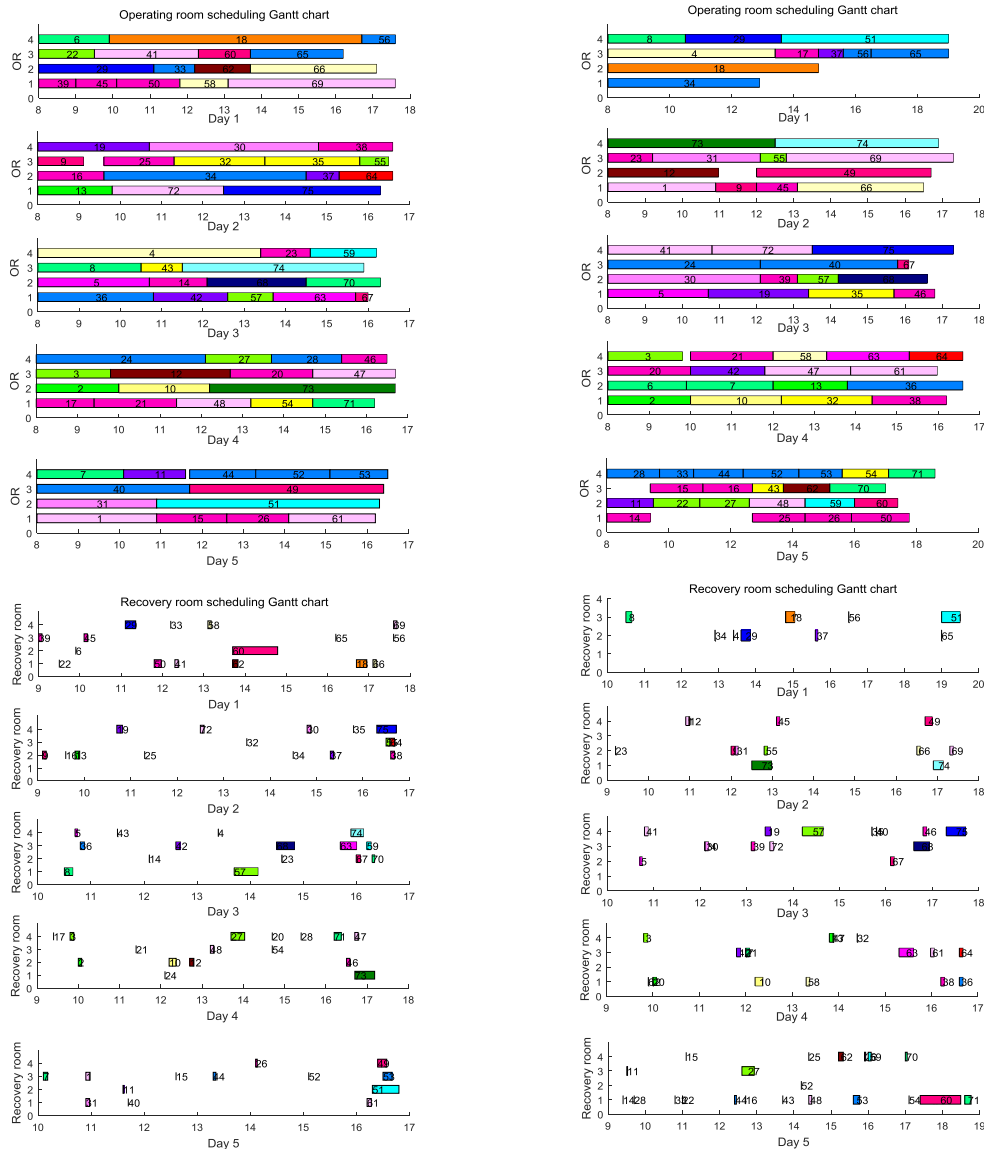
Figure 11. The Gantt chart for the small-size instance.

Table 5. Scheduling results for the small-size instance.

Day	OR	Patient ID	Starting time (operating room)	Leaving time (operating room)	Starting time (recovery room)	Leaving time (recovery room)
1	1	1	8:00	10:00	10:00	10:05
1	1	18	10:00	10:42	10:42	10:47
1	1	24	10:42	15:54	15:54	15:58
1	2	17	8:00	13:24	13:24	13:54
1	2	21	13:24	14:30	14:30	14:57
1	2	26	14:30	15:48	15:48	15:53
2	1	27	8:00	9:24	9:24	9:56
2	1	13	9:48	11:24	11:24	11:43
2	1	15	11:24	15:30	15:30	15:35
2	2	2	8:00	9:48	9:48	9:53
2	2	8	9:48	11:36	11:36	11:42
2	2	14	11:36	15:30	15:30	16:35
3	1	5	8:00	11:12	11:12	11:17
3	1	28	11:12	15:42	15:42	16:10
3	2	6	8:00	11:00	11:00	11:05
3	2	11	11:00	14:12	14:12	14:17
3	2	19	14:12	16:06	16:06	16:11
4	1	7	8:00	10:54	10:54	11:00
4	1	9	10:54	15:54	15:54	15:58
4	2	3	8:00	12:18	12:18	12:22
4	2	22	12:18	13:54	13:54	13:58
4	2	25	13:54	15:24	15:24	15:32
5	1	16	8:00	9:36	9:36	9:36
5	1	10	10:48	14:30	14:30	14:30
5	1	23	14:30	15:48	15:48	15:53
5	2	4	8:00	10:48	10:48	10:53
5	2	12	10:48	12:18	12:18	12:18
5	2	20	12:18	15:48	15:48	15:50

For the solution obtained by the proposed method, the value of F_2 equals 2.85; sum up $f_t (\forall t \in T)$, denoted as f , the value of f equals 943.86. For the solution obtained by Fei et al.' [21] method, the value of F_2 equals 4.72, the value of f equals 936.33. The comparison shows that, our method and Fei et al.' [21] method all obtained excellent scheduling result, the value of F_2 obtained by the proposed method is better than that obtained by Fei et al.' [21] method, but, the value of f obtained by the proposed method is worse than that obtained by Fei et al.' [21] method.

(2) The medium-size instance numerical experiment result



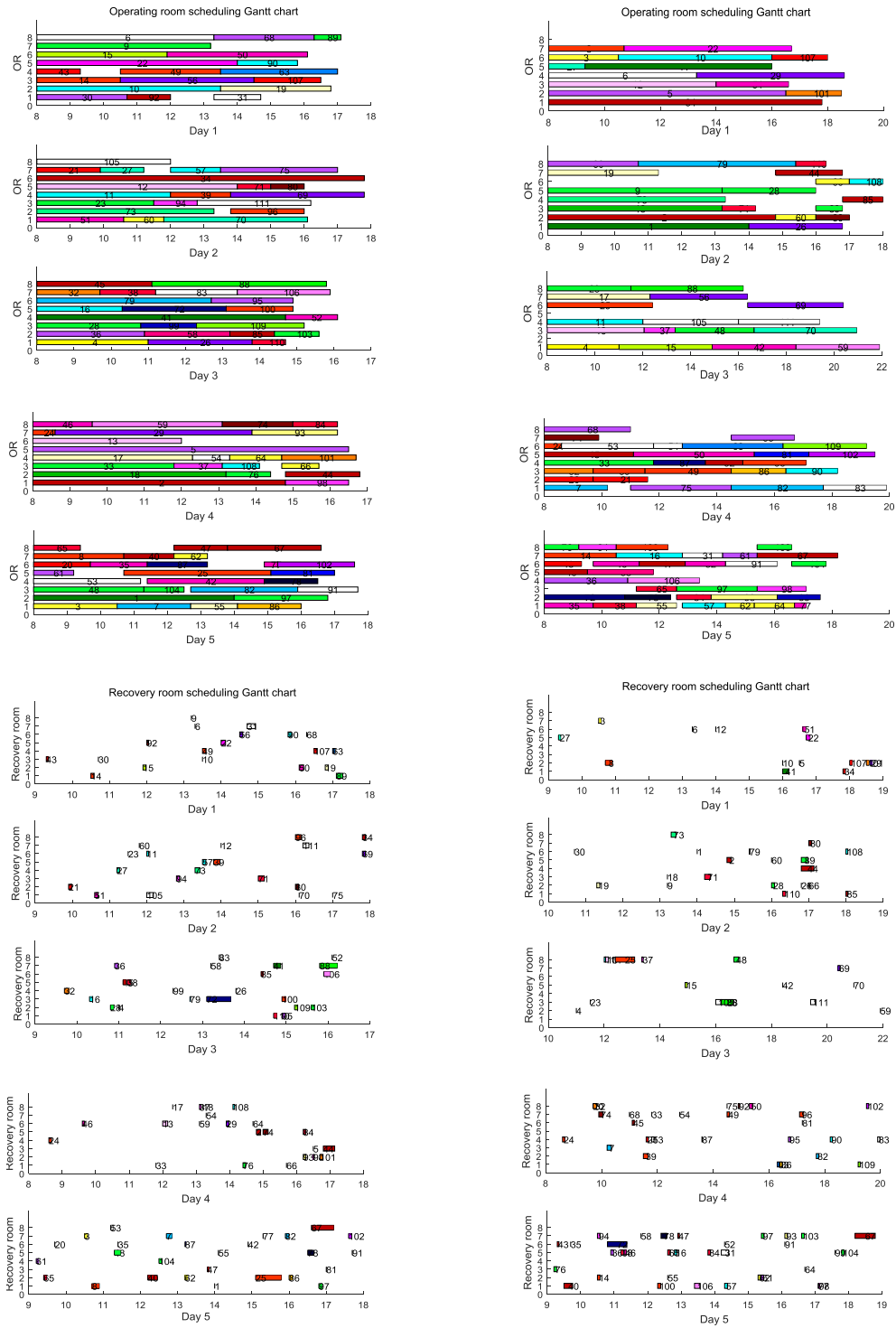
The solution obtained by the proposed method

The solution obtained by Fei et al.' [21] method

Figure 12. The Gantt chart for the medium-size instance.

For the solution obtained by the proposed method, the value of F_2 equals 15.85; the value of f equals 996.66. For the solution obtained by Fei et al.' [21] method, the value of F_2 equals 34.91, the value of f equals 1055.99. The comparison shows that, both the value of F_2 and the value of f obtained by the proposed method are better than those obtained by Fei et al.' [21] method.

(3) The large-size instance numerical experiment result



The solution obtained by the proposed method The solution obtained by Fei et al.' [21] method

Figure 13. The Gantt chart for the large-size instance.

For the solution obtained by the proposed method, the value of F_2 equals 46.80; the value of f equals 1018.15. For the solution obtained by Fei et al.' [21] method, the value of F_2 equals 129.89,

the value of f equals 1150.36. The comparison shows that, both the value of F_2 and the value of f obtained by the proposed method are better than those obtained by Fei et al.' [21] method.

4.2.2. Numerical experiment for practical example

The proposed method can take into account both patients' satisfaction and operating room costs, and obtains Pareto sets showed in Figure 14, which proved the diversity of solutions. We choose a solution from Pareto sets and get its scheduling Gantt chart, as shown in Figure 15. The scheduling results are presented in Table 6. Based on Pareto sets, hospital managers can balance patients' satisfaction and operating room costs to choose one of solutions as the final solution.

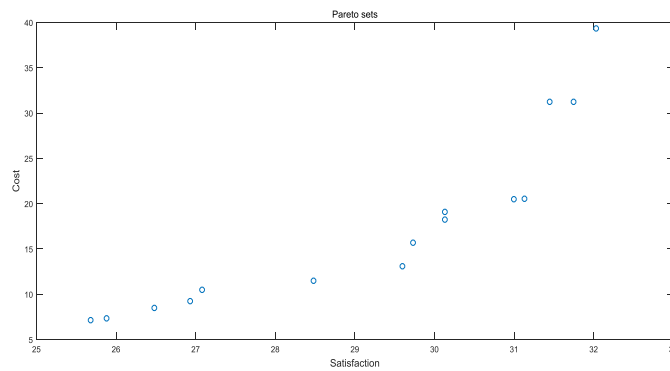


Figure 14. The Pareto set.

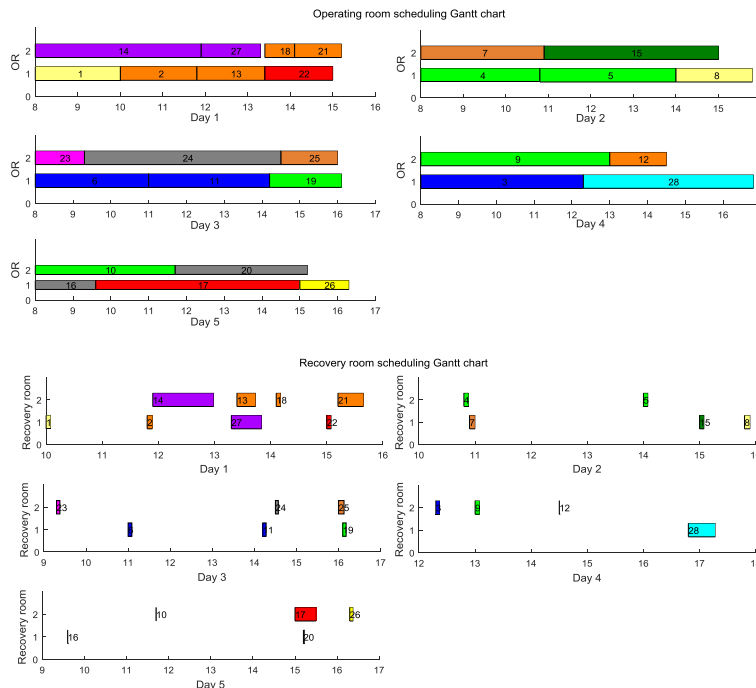


Figure 15. The scheduling Gantt chart of a solution.

Table 6. The scheduling results.

Day	OR	Patient ID	Starting time (operating room)	Leaving time (operating room)	Starting time (recovery room)	Leaving time (recovery room)
1	1	1	8:00	10:00	10:00	10:04
1	1	2	10:00	11:48	11:48	11:53
1	1	13	11:48	13:24	13:24	13:43
1	1	22	13:24	15:00	15:00	15:04
1	2	14	8:00	11:54	11:54	12:58
1	2	27	11:54	13:18	13:18	13:50
1	2	18	13:24	14:06	14:06	14:10
1	2	21	14:06	15:12	15:12	15:39
2	1	4	8:00	10:48	10:48	10:52
2	1	5	10:48	14:00	14:00	14:04
2	1	8	14:00	15:48	15:48	15:54
2	2	7	8:00	10:54	10:54	11:00
2	2	15	10:54	15:00	15:00	15:04
3	1	6	8:00	11:00	11:00	11:05
3	1	11	11:00	14:12	14:12	14:16
3	1	19	14:12	16:06	16:06	16:10
3	2	23	8:00	9:18	9:18	9:22
3	2	24	9:18	14:30	14:30	14:34
3	2	25	14:30	16:00	16:00	16:07
4	1	3	8:00	12:18	12:18	12:22
4	1	28	12:18	16:48	16:48	17:16
4	2	9	8:00	13:00	13:00	13:04
4	2	12	13:00	14:30	14:30	14:30
5	1	16	8:00	9:36	9:36	9:36
5	1	17	9:36	15:00	15:00	15:30
5	1	26	15:00	16:18	16:18	16:22
5	1	10	8:00	11:42	11:42	11:42
5	1	20	11:42	15:12	15:12	15:13

4.3. Discussions

In order to compare comprehensively, we consider not only F_2 and f , but also the total number of overtime hours in planning period (denoted as OT) and the total number of idle hours during the regular open period of operating rooms in planning period (denoted as IT). The comparison between the solution obtained by the proposed method and the solution obtained by Fei et al.' [21] method is shown in Table 7.

Table 7. The comparison results.

Instance	Method	F_2	f	OT (hours)	IT (hours)
The small-size instance	a	2.85	943.86	0.1	4.5
	b	4.72	936.33	0.2	4.6
The medium-size instance	a	15.85	996.66	10.5	1.6
	b	34.91	1055.99	21	12.1
The large-size instance	a	46.80	1018.15	17.6	32.6
	b	129.89	1150.36	60	75

a: The solution obtained by the proposed method. b: The solution obtained by Fei et al.' [21] method.

It can be seen from Table 7 that, for the small-size instance, the value of F_2 and OT obtained by the proposed method are better than those obtained by Fei et al.' [21] method, however, the value of f and IT obtained by Fei et al.' [21] method are better than those obtained by the proposed method. Hence, the proposed method is comparable to Fei et al.' [21] method. For the medium-size and large-size instances, the value of F_2 , f , OT and IT obtained by the proposed method are better than those obtained by Fei et al.' [21] method. Hence, the proposed method is better than Fei et al.' [21] method. To sum up, with the size increasing, the operating room scheduling solutions obtained by the proposed method have less idle time between two surgical cases, much higher utilization of operating rooms and produce less overtime.

The fact that the solution obtained by the proposed method is superior to that obtained by Fei et al. [21] method can be analysed from both the model and the algorithm.

In term of the model, our proposed method integrates the planning problem and the scheduling problem. For each allocation result obtained in the SCAP, there are n kinds of solutions in the SCSP. Therefore, when there are m kinds of allocation results in the SCAP, there are total $(m \times n)$ kinds of solutions. The solution space is huge. The method employed in Fei et al. [21] is a decomposition one, at first, in the SCAP, just determined an allocation result from m kinds of allocation result. Then, for this allocation result, there are n kinds of solutions in the scheduling layer. Therefore, the solution space is n . After comparison, it is obvious that the integrated method has a greater solution space and can find a better solution.

In term of the algorithm, a 2DPSO algorithm is proposed to solve this problem. In the 2DPSO algorithm, (1) parameters w , c_1 , c_2 are adjusted adaptively. w decreases linearly with the number of iterations increasing. While w is large, the particle keeps more historical information, leading to the diversity of swarm; on the contrary, while w is small, leading to the concentration of the swarm. c_1 (c_2) is adjusted adaptively according to the distance between the current particle and the local best (the global best), which balances the diversity and concentration of the swarm effectively; (2) crossover, insertion and exchange operators are designed to update the particle position, which balance the global exploration and local exploitation of the 2DPSO algorithm; (3) in the scheduling layer, the gap finding method is designed, This method pushes patients' operation time forward continuously. As long as the operating room is idle, the patient is inserted as early as possible, so that the patients' operation time can be arranged in a compact manner to achieve active scheduling. Hence, the utilization of operating rooms is increased. Moreover, as the size of instances increases, the optimization effect becomes more and more obvious.

5. Conclusions

The aim of this paper is to solve the joint problem of planning and scheduling patients in operating rooms on an operational level (weekly basis). We study how to optimize the operating rooms planning and scheduling problem simultaneously. A multi-objective optimization model is established. An integrated optimization approach is proposed and a group of experiments are carried out. The main conclusions are as follows.

(1) An integrated multi-objective integer programming model is proposed to plan and schedule the operating rooms in a hospital efficiently. The aim is to maximize the overall patients' satisfaction and minimize the cost of overtime in operating room as well as the daily cost of operating room and recovery beds.

(2) A bi-layer discrete particle swarm optimization algorithm is proposed to solve this problem. The Surgical Case Assignment Problem is modelled as a resource-constrained bin-packing problem and solved by discrete particle swarm optimization algorithm. The Surgical Case Sequencing Problem, based on the results obtained in the planning phase, is modelled as a two-stage hybrid flow-shop problem with additional consideration of surgeon's constraints and solved by discrete particle swarm optimization algorithm.

(3) The results of the numerical experiment show the proposed algorithm outperformed Fei et al.' [21] algorithm and can solve the operating room scheduling problem effectively and efficiently.

However, there are many limitations in our study. In term of the bi-layer discrete particle swarm optimization algorithm, the time computational complexity is high and the algorithm should be simplified. In term of the model of operating room scheduling problem, firstly, emergency cases are not taken into consideration and only the surgeries for the elective patients are considered; secondly, the constraints of some human resources and material resources are not taken into consideration, such as nurses, surgical equipment, etc.; finally, the preparation time for each operation and the clean-up time before leaving the operating room are included in the operating time. It increases the doctors' operating time. Hence, in the future, the bi-layer discrete particle swarm optimization algorithm should be further improved and the model considered should be more practical.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant (Grant No.51305024) and the Foundation of Shanxi Province (Grant No.2015KW-032).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. X. Chen, Z. P. Fan, Z. W. Li, X. L. Han, X. Zhang and H. C. Jia, A two-stage method for member selection of emergency medical service, *J. Comb. Optim.*, **30** (2015), 871–891.

2. B. Wang, X. B. Han, X. X. Zhang, and S. H. Zhang, Predictive-reactive scheduling for single surgical suite subject to random emergency surgery, *J. Comb. Optim.*, **30** (2015), 949–966.
3. M. E. Bruni, P. Beraldi, and D. Conforti, A stochastic programming approach for operating theatre scheduling under uncertainty, *Ima. J. Manage. Math.*, **26** (2015), 99–119.
4. Z. X. Zhao and X. P. Li, Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming, *Oper. Res. Health. Car.*, **3** (2014), 160–167.
5. S. Neyshabouri and B. P. Berg, Two-stage robust optimization approach to elective surgery and downstream capacity planning, *Eur. J. Oper. Res.*, **260** (2016), 21–40.
6. D. Min and Y. Yih, An elective surgery scheduling problem considering patient priority, *Comput. Oper. Res.*, **37** (2010), 1091–1099.
7. A. Jebali and A. Diabat, A Chance-constrained operating room planning with elective and emergency cases under downstream capacity constraints, *Comput. Ind. Eng.*, **114** (2017), 329–344.
8. M. Lamiri, X. L. Xie, A. Dolgui and F. Grimaud, A stochastic model for operating room planning with elective and emergency demand for surgery, *Eur. J. Oper. Res.*, **185** (2008), 1026–1037.
9. A. Abedini, H. H. Ye and W. Li, Operating room planning under surgery type and priority constraints, *Procedia. Manuf.*, **5** (2016), 15–25.
10. N. Dellaert and J. Jeunet, A variable neighborhood search algorithm for the surgery tactical planning problem, *Comput. Oper. Res.*, **84** (2017), 1–10.
11. Y. Yang, B. Shen, W. Gao, Y. Liu and L. W. Zhong, A surgical scheduling method considering surgeons' preferences, *J. Comb. Optim.*, **30** (2015), 1016–1026.
12. T. M. Range, R. M. Lusby and J. Larsen, A column generation approach for solving the patient admission scheduling problem, *Eur. J. Oper. Res.*, **235** (2014), 252–264.
13. W. Xiang, J. Yin and G. Lim, An ant colony optimization approach for solving an operating room surgery scheduling problem, *Comput. Ind. Eng.*, **85** (2015), 335–345.
14. A. Riise, C. Mannino and E. K. Burke, Modelling and solving generalised operational surgery scheduling problems, *Comput. Oper. Res.*, **66** (2016), 1–11.
15. D. Duma and R. Aringhieri, An online optimization approach for the real time management of operating rooms, *Oper. Res. Health. Car.*, **7** (2015), 40–51.
16. F. Guerriero and R. Guido, Operational research in the management of the operating theatre: A survey, *Health. Car. Manage. Sci.*, **14** (2011), 89–114.
17. E. Demeulemeester, J. Beliën, B. Cardoen and M. Samudra, Operating room planning and scheduling, *Eur. J. Oper. Res.*, **201** (2010), 921–932.
18. R. M. Hallah and A. H. Al-Roomi, The planning and scheduling of operating rooms: A simulation approach, *Comput. Ind. Eng.*, **78** (2014), 235–248.
19. R. Aringhieri, P. Landa, P. Soriano, E. Tànfani and A. Testi, A two level metaheuristic for the operating room scheduling and assignment problem, *Comput. Oper. Res.*, **54** (2015), 21–34.
20. R. Guido and D. Conforti, A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem, *Comput. Oper. Res.*, **87** (2016), 270–282.
21. H. Y. Fei, N. Meskens and C. B. Chu, A planning and scheduling problem for an operating theatre using an open scheduling strategy, *Comput. Ind. Eng.*, **58** (2010), 221–230.

22. J. M. Molina-Pariente, V. Fernandez-Viagas and J. M. Framinan, Integrated operating room planning and scheduling problem with assistant surgeon dependent surgery durations, *Comput. Ind. Eng.*, **82** (2015), 8–20.
23. P. Landa, R. Aringhieri, P. Soriano, E. T ànfani and A. Testi, A hybrid optimization algorithm for surgeries scheduling, *Oper. Res. Health. Car.*, **8** (2016), 103–114.
24. A. J. Fong, M. Smith and A. Langerman, Efficiency improvement in the operating room, *J. Surg. Res.*, **204** (2016), 371–383.
25. D. N. Pham and A. Klinkert, Surgical case scheduling as a generalized job shop scheduling problem, *Eur. J. Oper. Res.*, **185** (2008), 1011–1025.
26. L. W. Zhong, S. C. Luo, L. D. Wu, L. Xu, J. H. Yang and G. C. Tang, A two-stage approach for surgery scheduling, *J. Comb. Optim.*, **27** (2014), 545–556.
27. R. Burdett and E. Kozan, An integrated approach for scheduling health care activities in a hospital, *Eur. J. Oper. Res.*, **264** (2017), 756–773.
28. A. Abedini, W. Li and H. H. Ye, An optimization model for operating room scheduling to reduce blocking across the perioperative process, *Procedia. Manuf.*, **10** (2017), 60–70.
29. J. N. D. Gupta, Two-stage, hybrid flow shop scheduling problem, *J. Oper. Res. Soc.*, **39** (1988), 359–364.
30. J. Q. Li, H. Y. Sang, Y. Y. Han, C. G. Wang and K. Z. Gao, Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions, *J. Cleaner. Prod.*, **181** (2018), 584–598.
31. J. Q. Li, Q. K. Pan and M. F. Tasgetiren, A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, *Appl. Math. Model.*, **38** (2014), 1111–1132.
32. H. C. Liu, L. Gao and Q. K. Pan, A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem, *Exp. Syst. Appl.*, **38** (2011), 4348–4360.
33. J. Q. Li, Q. K. Pan and F. T. Wang, A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem, *Appl. Soft. Comput. J.*, **24** (2014), 63–77.
34. X. Y. Li and L. Gao, An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, *Int. J. Prod. Econ.*, **174** (2016), 93–110.
35. S. Gao and C. G. Cao, Convergence analysis of particle swarm optimization algorithm, *Sci. Technol. Eng.*, **4** (2008), 25–32.
36. M. A. Ghorbani, R. Kazempour, K. W. Chau and S. Shamshirband, Forecasting pan evaporation with an integrated Artificial Neural Network Quantum-behaved Particle Swarm Optimization model: A case study in Talesh, Northern Iran, *Eng. Appl. Comput. Fluid. Mech.*, **12** (2018), 724–737.
37. J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proc. IEEE. Int. Conf. Neural. Networks.*, **4** (1995), 1942–1948.
38. Q. K. Pan, W. H. Wang and J. Y. Zhu, Modified discrete particle swarm optimization algorithm for no-wait flow shop problem, *Comput. Integr. Manuf. Syst.*, **13** (2007), 1127–1130.
39. X. L. Wu, S. D. Sun, J. J. Yu and H. F. Zhang, Research on multi-objective optimization for flexible job shop scheduling, *Comput. Integr. Manuf. Syst.*, **12** (2006), 731–736.
40. M. J. Tessler, S. J. Kleiman and M. M. Huberman, A “zero tolerance for overtime” increases surgical per case costs, *Can. J. Anesth.*, **44** (1997), 1036–1041.

41. M. Schuster, T. Standl, J. A. Wagner and J. Berger, Effect of different cost drivers on cost per anesthesia minute in different anesthesia subspecialties, *Anesthesiology*, **101** (2004), 1435–1443.
42. Y. Shi and R. C. Eberhart, Empirical study of particle swarm optimization, *Proc. IEEE Congr. Evol. Comput.*, (1999), 1945–1950.
43. X. L. Wu and Y. J. Sun, A green scheduling algorithm for flexible job shop with energy-saving measures, *J. Cleaner. Prod.*, **172** (2017), 3249–3264.



AIMS Press

©2019 author name, licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)