# Dynamic three-stage operating room scheduling considering patient waiting time and surgical overtime costs

**Article** *in* Journal of Combinatorial Optimization · January 2020

DOI: 10.1007/s10878-019-00463-5



**5 authors**, including:

Shuwan Zhu
Hefei University of Technology


Wenjuan Fan
Hefei University of Technology


Panos Pardalos
University of Florida

Journal of Combinatorial Optimization (2020) 39:185–215
https://doi.org/10.1007/s10878-019-00463-5

# Dynamic three-stage operating room scheduling considering patient waiting time and surgical overtime costs

**Shuwan Zhu**[1,2] · **Wenjuan Fan**[1,2] · **Tongzhu Liu**[3] · **Shanlin Yang**[1,2] · **Panos M. Pardalos**[4]

Published online: 29 October 2019


## Abstract

In this paper, we study a dynamic operating room scheduling problem which consists of three stages. The problem simultaneously tackles the capacity allocation of operating rooms to each specialty, assignment of operating rooms to surgeons, assignment and sequence of patients. To lower the total costs of operating rooms from both sides of patients and operating rooms, a mathematical model is proposed with objective of minimizing the patient waiting costs and operating room overtime costs. Some structural properties of the studied problem are proposed, and two heuristic algorithms are presented to solve the patient assignment problem based on these structural properties. The studied operating room scheduling problem is proved to be NP-hard, and a hybrid GWO-VNS algorithm combining Grey Wolf Optimizer (GWO) with Variable Neighbourhood Search (VNS) is developed to obtain a good solution, where the heuristic algorithms are incorporated. Finally, computational experiments are conducted to test the efficiency, stability, and convergence speed of the proposed algorithm and compared with other mainstream algorithms. The results show that our proposed algorithm outperforms the compared algorithms.

**Keywords** Operating room scheduling · Hybrid algorithm · GWO-VNS · Heuristic

✉ Wenjuan Fan
fanwenjuan@hfut.edu.cn

1 School of Management, Hefei University of Technology, Hefei, China

2 Key Laboratory of Process Optimization and Intelligent Decision-making of Ministry of Education, Hefei, China

3 The First Affiliated Hospital of University of Science and Technology of China, Hefei, China

4 Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

# 1 Introduction

In most hospitals, the department of operating rooms (ORs) is both a cost and revenue center. For OR managers, one of the main questions is how to lower the total costs of ORs from both sides of the patients and ORs. In term of patients, their satisfaction can be increased by decreasing the expenses. Motivated by this consideration, we try to solve a three-stage OR scheduling problem with the objective of minimizing the overall costs which consist of OR overtime expenses and patient hospitalization expenses. In this three-stage problem, we need to determine at the same time: (1) the OR days (which OR on which day) allocated to different specialties, (2) the OR days assigned to different surgeons and (3) the OR days allocated to different patients and the patient sequence in each OR day.

In most previous studied problems for multi-functional ORs, the number of ORs assigned to each surgical specialty weekly is assumed to be fixed. However, each specialty has different workload in different periods, since the demands for ORs by different specialties are changing due to many factors, such as the seasons, epidemic, etc. Thus, it is necessary to consider the OR scheduling problem from more decision levels to flexibly accommodate the dynamic demands. Motivated by this consideration, we make decisions on not only patient and surgeon scheduling, but also on the capacity allocation for each specialty simultaneously. It is helpful to respond to weekly fluctuations in demand on the level of specialties, considering the requirements of the whole waiting list. What is more, the decision-making of surgeon scheduling is related to the results of specialty scheduling, and the decision-making of patient scheduling is related to the results of surgeon scheduling, the correlation among these decisions in different levels is very strong. Therefore, different from previous studies which split the OR scheduling problem in three levels and solves each level separately, we jointly consider these three levels simultaneously other than separately since all decision levels strongly interact with each other. In our study, the capacity allocation problem addresses decision-making on how overall OR time is divided among surgical specialties. Then the Master Surgical Schedule (MSS), a cyclic timetable, is established to define the specific assignment of OR time to individual specialties and individual surgeons. The problem in the last stage, which is called "surgical case assignment and scheduling problem" is to assign a specific OR and a date to each patient over the planning horizon, and then to determine the sequence of patients.

To the best of our knowledge, the vast majority of papers only consider one of the stages in OR scheduling problem. Saadouli et al. (2015) choose patients to be scheduled in the selected day. Astaraky and Patrick (2015) and Holte and Mannino (2013) allocate available OR to each surgical specialty. Min and Yih (2010) assign surgical blocks to each patient. Hosseini and Taaffe (2015) use linear programming to solve the problem of allocating OR block time to each surgical group taking into account both over-utilized time and under-utilized time. Penn et al. (2017) assign blocks of time in specific OR to each surgeon. They propose a multiple criteria mixed-integer linear programming model that helps head nurses in building new MSS, considering reducing the maximum number of beds required, and surgeons' availability. However, there exist limited studies that address more than one stage in OR scheduling problem simultaneously. Lee and Yih (2014) determine the sequence of surgeries in the first stage and the definite starting times for all surgical cases in the second stage. Pham and Klinkert (2008) allocate hospital resources to individual patients and decide on the time to perform the surgeries. Aringhieri et al. (2015) address a joint problem including two stages, MSS and surgical case scheduling with the decisions of OR time allocation given as input data. Vancroonenburg et al. (2015) use a two-phase approach to address the OR scheduling problem also including two stages, MSS and surgical case scheduling. They develop a flexible decision support model for multi-day OR scheduling, considering human dependencies and material dependencies. The main shortcoming of this method is that the decisions made in the first two stages cannot be modified according to the feedback from the third stage. Thus, the interaction between them is ignored, and no trade-off is investigated.

Only few papers study the OR scheduling problems in three stages, which are quite rare in the literature. Testi et al. (2007) select the number of sessions to be scheduled for each ward in the first stage, assign wards and ORs in the second stage, and determine sequence of surgeries in the third stage. A discrete-event simulation model is used to evaluate the schedule. Guido and Conforti (2017) study on the OR time assigned to each surgical specialty, to each surgical team, and choose patients to be scheduled for each surgical team. An integer linear programming model is proposed with the aim of maximizing the number of scheduled patients. Our study covers decisions from three decision stages—capacity allocation stage, Master Surgery Schedule stage and operational stage. The comparisons between existing studies and our study are shown in Table 1.

Considering the complexity of operating room scheduling problems, heuristics and meta-heuristic algorithms are typically developed. Recently, Lin and Chou (2019) present a hybrid genetic algorithm that solves the problem of assigning a set of surgeries to several multifunctional ORs with the goal of minimizing the overtime-operating cost and the wasting cost for the under time. Qu et al. (2013) study the problem of designing a weekly scheduling template for clinics. The objective is to minimize patient waiting time, provider idle time, and provider overtime. A Monte Carlo sampling based genetic algorithm is developed and applied to real data from a real women's clinic. Marques et al. (2015) use a constructive and improvement heuristic approach to solve an elective surgery scheduling problem.

For our problem, we develop a novel hybrid GWO-VNS algorithm by combining the procedure and features of these two meta-heuristic algorithms. Grey Wolf Optimizer (GWO) is a population-based meta-heuristic optimization algorithm, first proposed by Van Houdenhoven et al. (2007). It is based on the hunting behavior and social hierarchy of grey wolves. The first three levels of leadership are respectively considered as the first, second and third best solutions to achieve the objective function. Their positions are closer to the prey, and other wolves are forced to update their positions. Compared with other swarm intelligence algorithms, its significant characteristics make is possible for it to be widely used in various optimization problems: it has very few parameters and does not need derivative information in the initial search. Cardoen et al. (2006) solve a two-stage assembly flow shop scheduling problem by GWO and showed the better performance of GWO over other well-known meta-heuristic algorithms. Mirjalili et al. (2014) propose the application of a hybrid GWO-GA algorithm for optimizing a dynamic welding scheduling problem. To the best

**Table 1** Comparisons among key-related work and this study

| Paper | Stage | Objective | Algorithm | Analysis | Resource | Complexity |
|---|---|---|---|---|---|---|
| Riise et al. (2016) | 3 | Minimize makespan | Heuristic | A | Surgeons, ORs, recovery units | NP-hard |
| Vijayakumar et al. (2013) | 3 | Minimize costs associated with resources | Heuristic | A | ORs, surgeons | NP-hard |
| Denton et al. (2010) | 3 | Minimize the amount by which bin capacity is exceeded | Heuristic | A | ORs, surgeons, nurses | NP-hard |
| Aringhieri et al. (2015) | 2, 3 | A cost function | Heuristic | A | ORs | NP-hard |
| Vancroonenburg et al. (2015) | 2, 3 | Minimize the total surgical duration | Heuristic | A | ORs, surgeons | – |
| Choi and Wilhelm (2014) | 2, 3 | Maximize excess revenue | Dynamic programming | E | ORs, surgeons | – |
| Guinet and Chaabane (2003) | 2, 3 | A cost function | Heuristic | A | ORs, surgeons, equipment | NP-hard |
| Guido and Conforti (2017) | 1, 2, 3 | Maximizes the number of scheduled patients | Heuristic | A | ORs, surgeons | – |
| Tànfani and Testi(2010) | 1, 2, 3 | Minimize sum of weighted times of all patients | Heuristic | A | ORs, ICU beds | NP-hard |
| Testi et al. (2007) | 1, 2, 3 | Maximize the sum of the benefit of each session | Simulation model | E | ORs, surgeons | NP-hard |
| Our study | 1, 2, 3 | A cost function | Heuristics and GWO-VNS | A | ORs, surgeons | NP-hard |

Stage: 1—assign specific OR time to each specialty, 2—assign OR time to each surgeon, 3—assign a surgery date and an OR to each patient. Analysis: A—approximate, E—exact

of the authors' knowledge, GWO has not been extended to the surgical scheduling problem so far. Variable neighborhood search (VNS) is a local search meta-heuristic, first proposed by Hansen and Mladenović (2001). Improved VNS has been proved to be effective when applied to OR planning problem. Jebali et al. (2006) address the problem of scheduling a set of elective surgery patients into multiple ORs. They

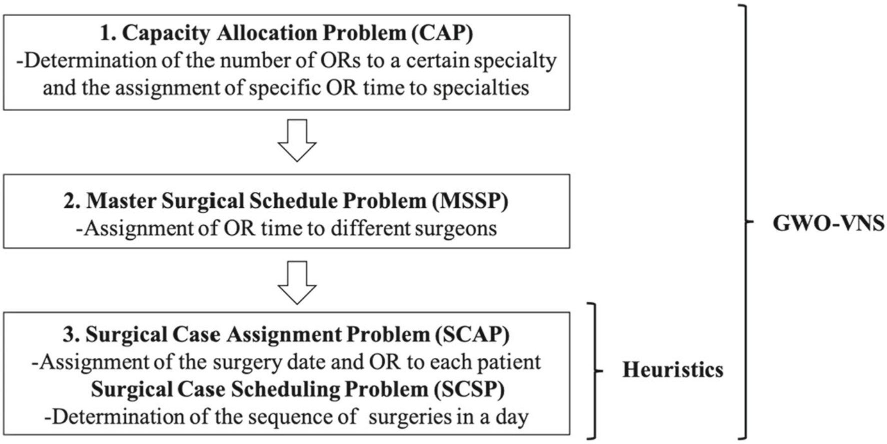



**Fig. 1** The methods for our integrated problem in three stages

propose a general solution framework taking advantage of the flexibility of VNS. Lei and Guo (2016) formulate a model for the scheduling problem for the treatments of resident patients in hospital for a given day and proposed Reduced VNS to solve it.

In this paper, we focus on a realistic integrated OR scheduling problem, which includes the OR days assigned to each surgical specialty in the first stage, the OR days assigned to each surgeon in the second stage, and the subsets of elective patients to be operated in each OR day in the third stage. Some key structural properties are first identified, and two heuristic algorithms are proposed. We use a hybrid GWO-VNS algorithm combining Grey Wolf Optimizer (GWO) with Variable Neighbourhood Search (VNS) to solve the studied problem, where the heuristic algorithms are incorporated to determine the assignment of patients. Figure 1 demonstrates the methods for our integrated problem in three stages.

Distinct from previous studies, the contributions are as follows: (1) We simultaneously consider the three decision stages mentioned above instead of taking account of them separately in three consecutive phases, featured by patients' surgical risk coefficients. Since the decision-making of the latter stage is based on the decision-making results of the former stages, the correlation among these stages is very strong. And thus, we jointly consider these three stages other than separately. It is helpful to respond to weekly fluctuations in demand for ORs by different specialties, considering the whole waiting list. (2) After the NP-hardness of our problem is proved, some key structural properties are first identified, and two heuristic algorithms are proposed for patient scheduling problem. (3) Based on the derived structural properties and the heuristic algorithms for the patient scheduling problem, an effective hybrid GWO-VNS algorithm is proposed to solve the operating room problem.

The remainder of the paper is organized as follows. Section 2 provides a detailed list of the objective and constraints that constitute the combinatorial optimization problem. In Sect. 3, some structural properties and scheduling rules of the proposed problem are derived, and two heuristic algorithms are developed to determine the OR days of surgical cases and their sequence. In Sect. 4, a hybrid GWO-VNS algorithm incorporating the proposed heuristic algorithms is developed to solve the integrated problem. The effectiveness and stability of our algorithm are reported and discussed in Sect. 5 through computer comparative experiments. Section 6 concludes the important findings and puts forward some future research directions.

## 2 Problem description and model

### 2.1 Problem description

This paper focuses on a realistic OR scheduling problem under a block scheduling strategy, proved to be NP-hard. More specifically, we solve the Capacity Allocation Problem (CAP) in the first stage to assign OR capacity to each specialty in the planning horizon. In the second stage, we solve the Master Surgical Schedule Problem (MSSP) to allocate OR days to each surgeon. In the third stage, we solve the Surgical Case Assignment Problem (SCAP) in which OR days are assigned to patients and Surgical Case Scheduling Problem (SCSP) in which the sequence is determined for the patients in each OR day. The resource constraints considered include the availability, and the number and type of ORs and surgeons. Figure 2 is the diagram of the problems to be solved in different stages.

In the first stage, the problem is to determine the number of ORs allocated to each specialty in each day during the one-week planning horizon. The resource composition of $r$ room on $d$ day is denoted as $(r, d)$, and we call it an OR day $(r, d)$. In the second stage, the problem is to determine the OR days allocated to each surgeon by specialty given the results of CAP. After CAP and MSS, each surgeon has several OR days to schedule his/her list of patients, and then in the third stage the problem is to allocate OR days to all patients in the waiting list and determine the surgery sequence in each $(r, d)$. Each patient's surgery $i$ (here we assume that each surgery is operated only once) has an estimated duration and a surgical risk coefficient, evaluated by the attending surgeon for scheduling. Larger coefficient indicates higher urgency of the surgery. In this context, OR resource is regarded as bottleneck resources and it inevitably leads to overtime in most cases.

For the three-stage problem, we make the following assumptions: (1) All patients in the waiting list have been in hospital before the decision day, and they should all be operated within the planning horizon. (2) The planning horizon is a week. (3) Each patient has been allocated to the corresponding surgeon. (4) Each surgeon has three OR days in a week, which is a realistic case.

### 2.2 Model

The notations used in this section are given in Table 2.

Fewer waiting days result in lower hospitalization costs and lead to higher satisfaction of patients. While scheduling most patients in early days leads to OR overtime costs. The trade-off between the patient waiting costs and the OR overtime costs must be resolved when determining the optimal OR schedule. Therefore, we consider an

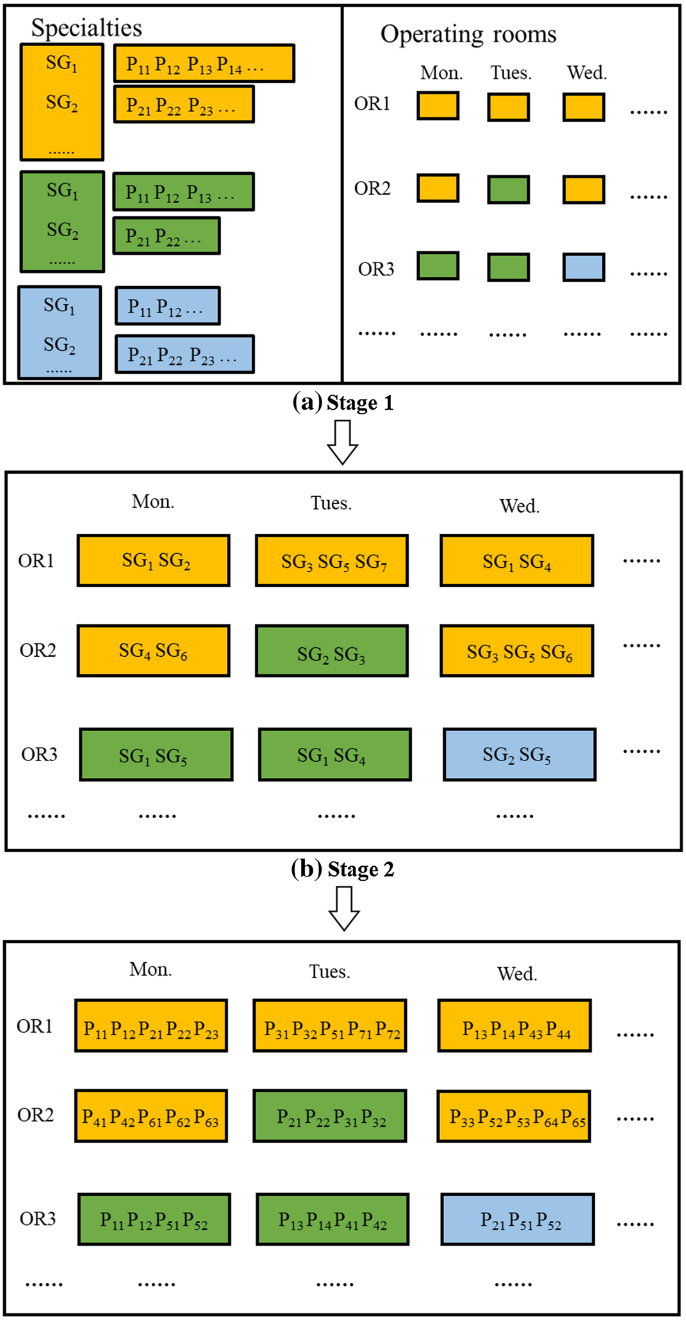



**Fig. 2** The diagram of the problems to be solved in different stages

**Table 2** Notations

| | |
|---|---|
| Sets | |
| $S$ | Set of surgeons |
| $I$ | Set of patients |
| $R$ | Set of ORs |
| $C$ | Set of specialties |
| $D$ | Set of weekdays |
| Indices | |
| $c$ | Index of specialties, $c \in$ C |
| $d$ | Index of days, $d \in$ D |
| $i$ | Index of patients, $i \in$ I |
| $r$ | Index of ORs, $r \in$ R |
| $s$ | Index of surgeons, $s \in$ S |
| Parameters | |
| $p_i$ | Surgery duration of $i$ patient |
| $w_i$ | Surgical risk coefficient evaluated by the attending surgeon for $i$ patient |
| $\tau_{c,s} = 1$ | If surgeon $s$ can do surgeries of $c$ specialty, 0 otherwise |
| $\theta_{i,s} = 1$ | If the attending surgeon of $i$ patient is $s$, 0 otherwise |
| $U$ | Regular opening hours of each OR |
| $O_{\max}$ | The max overtime of each OR |
| $C_{hos}$ | Hospitalization cost per day for each patient |
| $C_{over}$ | Overtime cost per hour for each OR |
| $E$ | The maximum number of surgeons in each OR |
| $d_i$ | The number of waiting days of $i$ patient |
| Decision variables | |
| $x_{c,r,d} = 1$ | If surgeries of $c$ specialty are assigned to ($r$, $d$), 0 otherwise |
| $y_{s,r,d} = 1$ | If $s$ surgeon is assigned to ($r$, $d$), 0 otherwise |
| $z_{i,d} = 1$ | If $i$ patient is assigned on $d$ day, 0 otherwise |
| $\gamma_{i,j} = 1$ | If $i$ patient precedes $j$ patient in the same ($r$, $d$), 0 otherwise |

objective function with two parts: the first part is the total hospitalization costs of all patients, and the second part is the total overtime costs of all ORs.

Objective :

$$\min\left\{\sum_{i\in I,s\in S,d\in D} z_{i,d}\theta_{i,s}d_i w_i C_{hos} + \sum_{d\in D,r\in R}\left[\max\left(0, \sum_{i\in I,s\in S} y_{s,r,d}z_{i,d}\theta_{i,s}p_i - U\right)\right]C_{over}\right\} \tag{1}$$

Subject to:

$$\sum_{c\in C,s\in S,d\in D} z_{i,d}\tau_{c,s}\theta_{i,s} = 1 \quad \forall i \in I \tag{2}$$

$$y_{s,r,d} z_{i,d} \tau_{c,s} \theta_{i,s} \le x_{c,r,d} \quad \forall i, d, r, c, s \tag{3}$$

$$\sum_{c \in C} x_{c,r,d} = 1 \quad \forall r, d \tag{4}$$

$$\sum_{r \in R} y_{s,r,d} \le 1 \quad \forall d, s \tag{5}$$

$$\sum_{i \in I, s \in S} y_{s,r,d} z_{i,d} \theta_{i,s} p_i \le U + O_{\max} \quad \forall d, r \tag{6}$$

$$\sum_{s \in S} y_{s,r,d} \le E \quad \forall rd \tag{7}$$

$$\gamma_{i,j} + \gamma_{j,i} \le 1 \quad \forall i, j \in I \tag{8}$$

$$\gamma_{i,h} \ge \gamma_{i,j} + \gamma_{j,h} - 1 \quad \forall i, j, h \in I \tag{9}$$

$$x_{c,r,d} \in \{0, 1\} \quad \forall c, r, d \tag{10}$$

$$y_{s,r,d} \in \{0, 1\} \quad \forall s, r, d \tag{11}$$

$$z_{i,d} \in \{0, 1\} \quad \forall i, d \tag{12}$$

$$\gamma_{i,j} \in \{0, 1\} \quad \forall i, j \tag{13}$$

Constraint (2) guarantees that all patients must be operated within the planning horizon. At the same time, it specifies that each patient should be assigned one time. Constraint (3) ensures that patients are assigned to rooms by specialty. Constraint (4) prevents assigning more than one specialty to each OR day. Constraint (5) ensures that surgeons do not overlap between rooms in the same day. Constraint (6) guarantees that the total operating time assigned to each OR day cannot exceed its general opening time plus overtime. Constraint (7) ensures that the number of surgeons in each OR day cannot be larger than the maximum number of surgeries. $E$ represents the maximum number of surgeries that could be completed in the maximum opening OR time each day. $E = (U + O_{\max})/\overline{p}_{c_{\min}}$, where $\overline{p}_{c_{\min}}$ denotes the minimum of the average duration of the surgeries among all specialties. Constraint (8) ensures that only one of $\gamma_{i,j}$ or $\gamma_{j,i}$ can be 1. Constraint (9) is required to maintain consistency of schedule between any three continuous surgeries in the same OR, such that if $i$ precedes $j$ and $j$ precedes $h$, then $i$ should precede $h$. Constraints (10–13) limit variable domains.

### 2.3 Complexity analysis

When the objective of the SCAP is to minimize resource-related costs, it becomes similar to the classical bin-packing problem with additional side constraints (Van Houdenhoven et al. 2007), known as NP-hard (Vijayakumar et al. 2013; Cardoen et al.

**Table 3** The bin-packing problem and case scheduling problem

| | Classical bin-packing problem | SCAP |
|---|---|---|
| Given | Items | Surgical cases |
| Variable | Bins | OR days |
| Objective | Minimum costs related to bins to allocate items | Minimum costs related to resources consumed by surgical cases |

2006). We list a more detailed comparison of the variants to illustrate that SCAP can be reduced to the bin-packing problem in Table 3.

Here we propose a clear complexity analysis to prove that our SCAP is NP-hard in the case of the block scheduling model through a reduction to 0–1 Multiple Knapsack problem (MKP). We consider a particular case with the following characteristics: $C = \{1\}$, $R = \{1\}$, $S = \{1\}$, and the mathematical formulation (2)–(13) can be simplified as follows: constraints (3) (4) can be removed due to $C = \{1\}$; constraint (5) can be omitted due to $R = \{1\}$; constraint (7) can be omitted due to $S = \{1\}$; constraint (10–11) are unnecessary. Constraint (8–9) (13) are related to the sequence of patients in each $(r, d)$. Here, we only consider the assignment of patients, and thus constraint (8–9) (13) can be removed. In addition, index $r$ can be omitted due to $R = \{1\}$. Consequently, the model can be rewritten as:

$$\min F = \left\{ \sum_{i\in I, d\in D} z_{i,d} d_i w_i C_{hos} + \sum_{d\in D} \left[ \left[ \max\left(0, \sum_{i\in I, s\in S} z_{i,d} p_i - U\right) \right] C_{over} \right] \right\}$$

$$\begin{aligned} s.t.\ & \sum_{d\in D} z_{i,d} = 1 \quad \forall i \\ & \sum_{i\in I} z_{i,d} p_i \le U + O_{\max} \quad \forall d \\ & z_{i,d} \in \{0, 1\} \quad \forall i, d \end{aligned}$$

Let $F^W = \sum_{i\in I} 5 w_i C_{hos} + \left(\sum_{i\in I, d=5} z_{i,d} p_i - U\right) C_{over}$ denote the value of the worst case, where the solution is that all patients are scheduled in the last day of the planning horizon. It is known that $F^W$ is a constant. $F = F^W - F'$, where $F'$ represents the contribution to the overall costs by the surgical risk coefficient $w_i$ multiplies the number of days in the waiting list that can be removed due to the decision of assigning the patients on day $t$. Thus, we can rewrite the model as:

$$\begin{aligned} \max\ F' = & \sum_{i\in I, d\in D} 5 z_{i,d} w_i C_{hos} \\ s.t.\ & \sum_{d\in D} z_{i,d} = 1 \quad \forall i \\ & \sum_{i\in I} z_{i,d} p_i \le U + O_{\max} \quad \forall d \\ & z_{i,d} \in \{0, 1\} \quad \forall i, d \end{aligned}$$

The above model can be regarded as a 0–1 MKP, which is NP-hard. Therefore, the NP-hardness of our problem is proved.

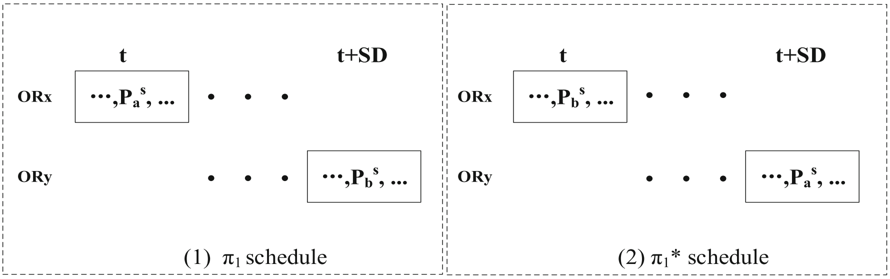



**Fig. 3** Different schedules in Lemma 1

## 3 The structural properties and the heuristic algorithms

In this paper, we propose a hybrid GWO-VNS algorithm (we leave it in Sect. 4) to solve the studied problem, where Heuristic Algorithms 1&2 are incorporated to obtain the patient assignment results, which is described in this section. There are two rounds of assignment to solve the patient assignment problem in the third stage. Specifically, in the first round of assignment, some patients in the waiting list are selected to be scheduled within the regular opening time of ORs. Some structural properties are derived and Lemma 1 is proposed, based on which Heuristic Algorithm 1 for the first round of assignment is obtained. In the second round of assignment, the remaining patients in the waiting list are scheduled to overtime, and Lemma 2 is derived. Based on Lemma 2, Heuristic Algorithm 2 for the second round of assignment is derived. It should be noted that if all patients can be scheduled within regular opening time of OR, the second round is not necessary.

### 3.1 First round of assignment

**Lemma 1** *When all surgeons' operating time does not occupy the overtime, the patient of each surgeon should be scheduled by his/her surgical risk coefficient, i.e., the higher the surgical risk coefficient is, the earlier the patient to be scheduled.*

***Proof*** Suppose $P_a^s$ and $P_b^s$ are two patients of surgeon $s$. Here we assume that $\pi_1$ is an optimal schedule in which $P_a^s$ is scheduled on day $t$, and $P_b^s$ is scheduled on day $(t + SD)$ $(SD > 0)$, $\left(w_a^s < w_b^s\right)$. Swap the position of $P_a^s$ and $P_b^s$ in the $\pi_1$ and the new scheduled is denoted as $\pi_1^*$ schedule (see Fig. 3).

The objective function of $\pi_1$ is $F(\pi_1) = \sum_{i \in I \backslash p_a, p_b} \left[\sum_{s \in S, d \in D} \theta_{i,s} z_{i,d} d_i w_i C_{hos}\right] + t w_a^s C_{hos} + (t + SD) w_b^s C_{hos} + \sum_{d \in D, r \in R} \left[Max\left(0, \sum_{s \in S, i \in I} y_{s,r,d} z_{i,d} \theta_{i,s} p_i - U\right)\right] C_{over}$. The objective function of $\pi_1^*$ schedule is $F\left(\pi_1^*\right) = \sum_{i \in I \backslash p_a, p_b} \left[\sum_{s \in S, d \in D} \theta_{i,s} z_{i,d} d_i w_i C_{hos}\right] + (t + SD) w_a^s C_{hos} + t w_b^s C_{hos} + \sum_{d \in D, r \in R} \left[Max\left(0, \sum_{s \in S, i \in I} y_{s,r,d} z_{i,d} \theta_{i,s} p_i - U\right)\right] C_{over}$. Thus, it can be inferred that when $\sum_{d \in D, r \in R} \left[Max\left(0, \sum_{s \in S, i \in I} y_{s,r,d} z_{i,d} \theta_{i,s} p_i - U\right)\right] C_{over} = 0$ (all surgeons' operating time does not occupy the overtime), $F\left(\pi_1^*\right) - F(\pi_1) = SD \cdot \left(w_a^s - w_b^s\right) \cdot C_{hos}$. Since we assume that $w_a^s < w_b^s$, it can be derived that

$F(\pi_1^*) < F(\pi_1)$, which conflicts with our assumption. And thus, it can be inferred that $\pi_1^*$ schedule is better than $\pi_1$ schedule.

Based on the Lemma 1, the following Heuristic Algorithm 1 is designed to determine the First Round of Assignment of patients to OR days within the general opening time.

---

Heuristic Algorithm 1:

---

Input: The waiting list of patients; the duration and the surgical risk coefficient of each patient; the results of the OR days assigned to each surgeon.

Output: The assignment of patients within regular opening time.

1. Index all the patients of the surgeons for each OR day in descending order of their risk coefficients and thus new patient lists can be obtained.
2. Set $status_i$=1 for each patient $i$. Set $\varepsilon_{\mathrm{m}}= U$ for each OR day $m$.
3. for (*k=1; k<6; k++*)
4.     for (*m=k; m<5R+1; m+=5*)
5.         for $P_i^{bl_m}$ in the waiting list of OR day *m*
6.             if $status_i = 1$ and $p_i \le \varepsilon_{\mathrm{m}}$, then
7.                 $status_i \leftarrow 2,\ \varepsilon_{\mathrm{m}} \leftarrow \varepsilon_{\mathrm{m}} - p_i,\ P_i.block\_id \leftarrow m$

9.             end if
10.             Update the waiting list for each surgeon and for each OR day
11.         end for
12.     end for
13. end for

---

*(Notations: $status_i$ denotes the allocation status of patient i (indicating whether the patient has been assigned). It equals to 1 if patient i hasn't been assigned, 2 otherwise. $\varepsilon_m$ denotes the time gap between the complete time of the last surgery in OR day m and the general OR closing time. $P_i.block\_id$ denotes the index of the OR day assigned to patient i).*

The time complexity of step 1 is *O(n logn)*. The complexity of the remaining steps is no more than *O(n)*. Consequently, the complexity of Heuristic Algorithm 1 is *O(n logn)*.

### 3.2 Second round of assignment

**Lemma 2** *If there are remaining unscheduled patients after regular opening time has been allocated, then the unscheduled patients should be assigned to the overtime. The unscheduled patient $p_k^s$ with surgical risk $w_k^s$ can either be assigned to $(OR_x, t)$ or $(OR_y, t+SD)$, and there exists an optimized solution with different conditions shown in Table 4, given that First Round of Assignment has been completed.*

***Proof*** Given that surgeon $s$ has been assigned to $(OR_x, t)$ and $(OR_y, t+SD)$ (SD > 0), there exist two schedules for the patient $P_k^s$ (withuration $p_k$). We assume that $\pi_1$ is an optimized schedule where $P_k^s$ is assigned to $(OR_y, t+SD)$, while in $\pi_1^*$ schedule, $\mathrm{P}_{\mathrm{k}}^{\mathrm{s}}$ is assigned to $(OR_x, t)$. The constructed schedules are shown in Fig. 4.

**Table 4** Summary of the rules in Lemma 2

| Case | Condition | Conclusion | |
|---|---|---|---|
| 1 | $0 \le \varepsilon_{ORx,t} < \varepsilon_{ORy,t+SD} < p_k$ | If $w_k^s > \frac{\tau(\varepsilon_{ORy,t+SD}-\varepsilon_{ORx,t})}{SD}$, $\pi_1^*$ is an optimized schedule | If $w_k^s < \frac{\tau(\varepsilon_{ORy,t+SD}-\varepsilon_{ORx,t})}{SD}$, $\pi_1$ is an optimized schedule |
| | $0 \le \varepsilon_{ORx,t} = \varepsilon_{ORy,t+SD} < p_k$ | $\pi_1^*$ is an optimized schedule | |
| | $0 \le \varepsilon_{ORy,t+SD} < \varepsilon_{ORx,t} < p_k$ | $\pi_1$ is an optimized schedule | |
| 2 | $\varepsilon_{ORx,t} < 0 \le \varepsilon_{ORy,t+SD} < p_k$ | If $w_k^s > \frac{\tau\varepsilon_{ORy,t+SD}}{SD}$, $\pi_1^*$ is an optimized schedule | If $w_k^{sn} < \frac{\tau\varepsilon_{ORy,t+SD}}{SD}$, $\pi_1$ is an optimized schedule |
| 3 | $\varepsilon_{ORy,t+SD} < 0 \le \varepsilon_{ORx,t} < p_k$ | If $w_k^s > -\frac{\varepsilon_{ORx,t}\tau}{SD}$, $\pi_1^*$ is an optimized schedule | If $w_k^s < -\frac{\varepsilon_{ORx,t}\tau}{SD}$, $\pi_1$ is an optimized schedule |
| 4 | $\varepsilon_{ORx,t} < 0$ and $\varepsilon_{ORy,t+SD} < 0$ | $\pi_1^*$ is an optimized schedule | |

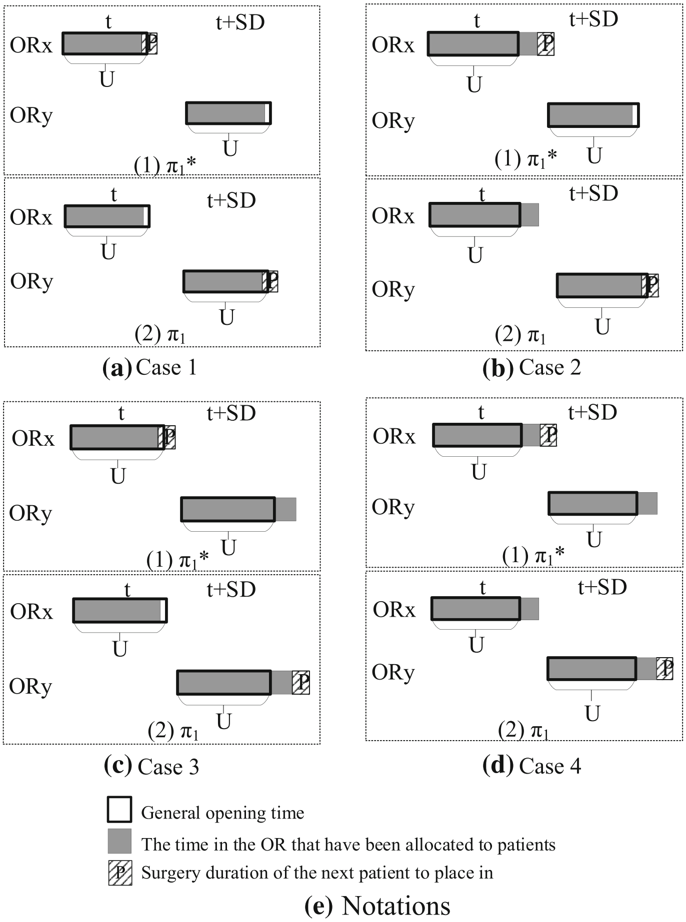



**Fig. 4** Different cases in Lemma 2

The first part of the objective function of $\pi_1^*$ and $\pi_1$ schedules are respectively $f_1(\pi_1^*) = \sum_{i\in I\setminus p_k, s\in S, d\in D} z_{i,d}\theta_{i,s}d_i w_i C_{hos} + t w_k^s C_{hos}$ and $f_1(\pi_1) = \sum_{i\in I\setminus p_k, s\in S, d\in D} z_{i,d}\theta_{i,s}d_i w_i C_{hos} + (t+SD)\cdot w_k^s C_{hos}$. Thus, it can be inferred that $F(\pi_1^*) - (\pi_1) = -SD\cdot w_k^s C_{hos} + f_2(\pi_1^*) - f_2(\pi_1)$.

Let $\varepsilon_{\mathrm{ORx},t}$ be the time gap between the completion time of the last surgery in $(OR_x, t)$ and the general OR closing time, i.e., $\varepsilon_{ORx,t} = U - \sum_{s\in S, i\in I} y_{s,ORx,t} z_{i,t}\theta_{i,s}p_i$. If $\varepsilon_{ORx,t} > 0$. It means the total time of surgeries in $(OR_x, t)$ is shorter than the general OR opening time. The value of $\tau$ equals to the overtime cost per hour for operating center divided by hospitalization cost per day for patient, i.e., $\tau = \frac{C_{over}}{C_{hos}}$. Let $A$ be the total costs of overtime of all OR days which has already happened before patient $P_k^s$ is assigned, i.e., $A = \sum_{d\in D, r\in R} \left[\max\left(0, \sum_{s\in S, i\in I\setminus p_k} y_{s,r,d} z_{i,d}\theta_{i,s}p_i - U\right)\right] C_{over}$.

(1) Case 1: $0 < \varepsilon_{ORx,t} < p_k$, $0 < \varepsilon_{ORy,t+SD} < p_k$

The second part of the objective function of $\pi_1^*$ and $\pi_1$ schedules are respectively $f_2(\pi_1^*) = A + (p_k - \varepsilon_{ORx,t}) + C_{over} + 0$ and $f_2(\pi_1) = A + 0 + (p_k - \varepsilon_{ORy,t+SD})C_{over}$. Thus, it can be inferred that $F(\pi_1^*) - F(\pi_1) = -SD \cdot w_k^s C_{hos} + A + (p_k - \varepsilon_{ORx,t})C_{over} - A - (p_k - \varepsilon_{ORy,t+SD})C_{over} = -SD \cdot w_k^s C_{hos} + (\varepsilon_{ORy,t+SD} - \varepsilon_{ORx,t})C_{over}$. Let $F(\pi_1^*) - F(\pi_1) < 0$, then $-SD \cdot w_k^s C_{hos} + (\varepsilon_{ORy,t+SD} - \varepsilon_{ORx,t})C_{over} < 0$, that is, when $\varepsilon_{ORx,t} = \varepsilon_{ORy,t+SD}$, $F(\pi_1^*) - F(\pi_1) = 0$; when $\varepsilon_{ORx,t} < \varepsilon_{ORy,t+SD}$, $w_k^s > \frac{\tau(\varepsilon_{ORy,t+SD} - \varepsilon_{ORx,t})}{SD}$, $F(\pi_1^*) - F(\pi_1) < 0$; when $\varepsilon_{ORx,t} > \varepsilon_{ORy,t+SD}$, $F(\pi_1^*) - F(\pi_1) < 0$.

(2) Case 2: $\varepsilon_{ORx,t} < 0 \leq \varepsilon_{ORy,t+SD} < p_k$.

The second part of the objective function of $\pi_1^*$ and $\pi_1$ schedules are respectively $f_2(\pi_1^*) = A + (p_k - \varepsilon_{ORx,t})C_{over} + 0$ and $f_2(\pi_1) = A + 0 + (p_k - \varepsilon_{ORy,t+SD})C_{over}$. Thus, it can be inferred that $F(\pi_1^*) - F(\pi_1) = -SD \cdot w_k^s C_{hos} + A + (p_k - \varepsilon_{ORx,t})C_{over} - A - (p_k - \varepsilon_{ORy,t+SD})C_{over} = -SD \cdot w_k^s C_{hos} + (\varepsilon_{ORy,t+SD} - \varepsilon_{ORx,t+SD})C_{over}$. Let s $F(\pi_1^*) - F(\pi_1) < 0$, then $-SD \cdot w_k^s C_{hos} + \varepsilon_{ORy,t+SD} C_{over} < 0$, that is, $w_k^s > \frac{\tau \varepsilon_{ORy,t+SD}}{SD}$.

(3) Case 3: $\varepsilon_{ORy,t+SD} < 0 \leq \varepsilon_{ORx,t} < p_k$

The second part of the objective function of $\pi_1^*$ and $\pi_1$ schedules are respectively $f_2(\pi_1^*) = A + (p_k - \varepsilon_{ORx,t})C_{over} + (-\varepsilon_{ORy,t+SD})C_{over}$ and $f_2(\pi_1) = A + 0 + (p_k - \varepsilon_{ORx,t})C_{over}$. Thus, it can be inferred that

$F(\pi_1^*) - F(\pi_1) = -SD \cdot w_k^s C_{hos} + A + (p_k - \varepsilon_{ORx,t})C_{over} + (-\varepsilon_{ORy,t+SD})C_{over} - A - (p_k - \varepsilon_{ORy,t+SD})C_{over} = -SD \cdot w_k^s C_{hos} - \varepsilon_{ORx,t} C_{over}$. Let $F(\pi_1^*) - F(\pi_1) < 0$, then $-SD \cdot w_k^s C_{hos} - \varepsilon_{ORx,t} C_{over} < 0$, that is, $w_k^s > -\frac{\varepsilon_{ORx,t}\tau}{SD}$.

(4) Case 4:$\varepsilon_{ORx,t} < 0$, $\varepsilon_{ORy,t+SD} < 0$.

The second part of the objective function of $\pi_1^*$ and $\pi_1$ schedules are respectively $f_2(\pi_1^*) = A + (p_k - \varepsilon_{ORx,t})C_{over} + (-\varepsilon_{ORy,t+SD})C_{over}$ and $f_2(\pi_1) = A + (-\varepsilon_{ORx,t})C_{over} + (p_k - \varepsilon_{ORy,t+SD})C_{over}$. Thus, it can be inferred that s $F(\pi_1^*) - F(\pi_1) = -SD \cdot w_k^s C_{hos} + A + (p_k - \varepsilon_{ORx,t})C_{over} + (-\varepsilon_{ORy,t+SD})C_{over} - A - (\varepsilon_{ORx,t})C_{over} - (p_k - \varepsilon_{ORy,t+SD})C_{over} = -SD \cdot w_k^s C_{hos}$. It is easily verified that $F(\pi_1^*) < F(\pi_1)$, which conflicts with our assumption. Thus, we can obtain the rule that when $\varepsilon_{ORx,t} < 0$, $\varepsilon_{ORy,t+SD} < 0$, $\pi_1^*$ is an optimized schedule.

Based on the Lemma 2, we propose a Heuristic Algorithm 2 to solve the Second Round of Assignment of patients to OR days in the overtime as follows:

Heuristic Algorithm 2:

Input: The assignment of OR days to surgeons; the number, type and duration of each patient's surgery; the scheduling results after executing Heuristic Algorithm 1.

Output: The assignment of patients in the overtime.

1. Calculate the values of $\varepsilon_{(OR_S,t_S)_1}$, $\varepsilon_{(OR_S,t_S)_2}$, $\varepsilon_{(OR_S,t_S)_3}$.
2. Sequence the patients whose $status_i = 1$ by the decreasing order of $w_i$.
3. for $P_i$ in the updated the total waiting list
4.     if $status_i$ =1, then
5.         Place $P_i$ into the OR day selected from $(OR_S, t_S)_1$, $(OR_S, t_S)_2$, $(OR_S, t_S)_3$ according to heuristic rules in Lemma 2.
6.         Update the values of $\varepsilon_{(OR_S,t_S)_1}$, $\varepsilon_{(OR_S,t_S)_2}$, $\varepsilon_{(OR_S,t_S)_3}$.
7.         $status_i \leftarrow 2$.
8.     end if
9. end for

*(Notations: $status_i = 1$ denotes the allocation status of patient i (indicating whether the patient has been assigned). It equals to 1 if patient i hasn't been assigned, 2 otherwise. $\varepsilon_{(OR_S,t_S)_1}$, $\varepsilon_{(OR_S,t_S)_2}$, $\varepsilon_{(OR_S,t_S)_3}$ respectively denotes the time gap between the complete time of the last surgery and the general OR closing time in the three OR days of surgeon s.*

The complexity of step 2 is $O(n^2)$. The complexity of the remaining steps is no more than *O(n)*. Consequently, the complexity of Heuristic Algorithm 2 is $O(n^2)$.

## 4 Metaheuristic-based hybrid approach

In this section, the hybrid GWO-VNS algorithm incorporating Heuristic Algorithm 1&2 is proposed for solving the problem. The key procedures of the proposed GWO-VNS are as follows:

### 4.1 Coding scheme

In order to code solution vector, we should consider the problem with the following three stages: (i) assign OR days to specialties (ii) assign OR days to surgeons (iii) assign OR days to patients.

Therefore, we find a solution for problems in these three stages. That is to generate an array of which the length is equal to the total number of all surgeons' OR days assigned in a week. We combine OR and day in the coding scheme to simplify the search process and make it more effective (see Fig. 5). For example, value *8* stands for OR day *(2,3)*. The solution can be denoted as $X = \{x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, \ldots, x_1^S, x_2^S, x_3^S\}$ and the three position values $x_1^S, x_2^S, x_3^S$ represent the three OR days assigned to surgeon.

### 4.2 Encoding correction strategy

In the iterative processes, infeasible solutions may be generated for six reasons: (1) OR days should be encoded with integers while search operators may generate decimals.

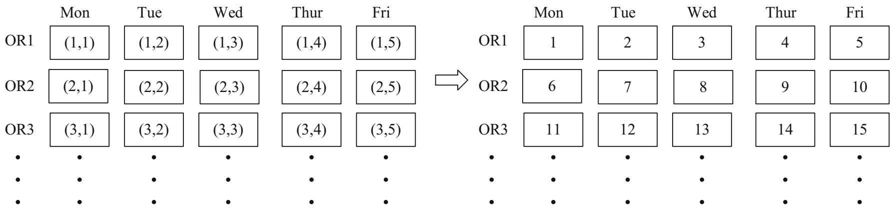



**Fig. 5** Position value for each (*r, d*)

We adopt the coding correction strategy which takes integer approximate values. (2) All position values should be in the range of *[1, 5R]*. The numbers in $X$ that are less than "1" are set to "1", and those are greater than "$5R$" are set to "$5R$". (3) There should not be any duplicate values at the positions for the same surgeon. If this occurs, leave one of the duplicate values as it is, and regenerate others. (4) Surgeons of different specialties should not be assigned in the same OR day. If this occurs, randomly select one surgeon $s$ from OR day $m$ and set his/her specialty as the specialty of the OR day $m$. Then shift surgeons of other specialties to other OR days by specialty. (5) Each surgeon should not be assigned to different ORs in a day. In the subset $\{x_1^s, x_2^s, x_3^s x_1^s$ } for surgeon $s$, if there exist any two position values whose difference is an integer multiple of five, leave one of the values as it is, and regenerate the other. (6) The number of surgeons in each OR day should not be larger than $E$ (defined in Sect. 2.2). If this occurs, regenerate the values at the positions for the surgeon.

### 4.3 VNS-based local search

In the following, we present several neighborhood structures $N_s$(s = 1, 2, 3, 4) (see Fig. 6) used within VNS and we propose a VNS-based local search procedure for improving the effectiveness of the traditional GWO.

(1) Operator 1: Transfer surgeon

In the operator, one surgeon selected from $(OR_x, d)$ is shifted to $(OR_y, d + SD)$. We should note that before the shift, the selected surgeon from $(OR_x, d)$ should not have been assigned to $(d + SD)$ day. Also, $(OR_x, d)$ and $(OR_y, d + SD)$ should be for the same specialty, or when $(OR_y, d + SD)$ is not occupied, the transfer can be done.

(2) Operator 2: Swap surgeons in the same day

In this operator, two surgeons selected respectively from $(OR_x, d)$, $(OR_y, d)$, are swapped with each other. We should note that before the swap, $(OR_x, d)$ and $(OR_y, d)$ should be of the same specialty.

(3) Operator 3: Swap two surgeons in different days

In this operator, two surgeons selected respectively from $(OR_y, d)$, $(OR_y, d + SD)$ are swapped with each other. We should note that before the swap, the selected surgeon in $(OR_x, d)$ should not have been assigned to ($d$ + SD) day, and the selected surgeon in $(OR_y, d + SD)$ should not have been assigned to $d$ day. Also, $(OR_x, d)$ and $(OR_y, d + SD)$ should be for the same specialty.

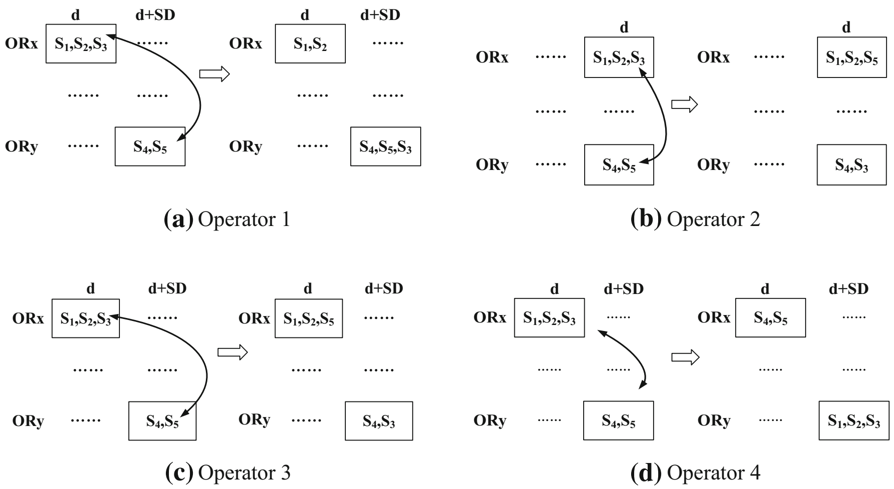



**Fig. 6** Local search operators

(4) Operator 4: Swap surgeons in different OR days

In the swap, all the surgeons in $(OR_x, d)$ and $(OR_y, d + SD)$ are swapped. We should note that before the swap, the surgeons in $(OR_x, d)$ should not have been assigned to $(d$ + SD) day, and the surgeons in $(OR_y, d + SD)$ should not have been assigned to $d$ day.

The VNS-based local search procedures for improving the effectiveness of the traditional GWO is described as follows:

**VNS-based local search procedures:**

Get an initial solution $X_0$

Set $s = 1$. $e = 1$

While ($s \leq$ 4), do

  Shake $X_0$ and obtain a feasible solution $X_0{}'$ $(X_0{}' \in N_s(X_0))$

  Execute local search from $X_0{}'$ until a local optimal solution $X_0{}''$ is reached

  If $X_0{}''$ is better than $X_0$ , then

    $X_0 = X_0{}''$, $s$=1

  Else

    $s$ = s+1

End while

*(Notations: the four neighborhood structures $N_s$ (s=1,2,3,4) are as shown in Figure 6.)*

The details of the operation of "local search" in the VNS-based local search are as follows:

| **The details of the local search in the VNS-based local search procedures:** | |
|---|---|
| 1. | Get a solution $X$ |
| 2. | For (*e=0; e<E; e++*) |
| 3. | $X' \in N_s(X)$ |
| 4. | If $X'$ is better than $X$ , then |
| 5. | $X = X'$ |
| 6. | End if |
| 7. | End for |

*(Notations: E denotes iteration number of local search.)*

### 4.4 Framework of GWO-VNS

The main steps and framework of GWO-VNS are described as shown in Table 5 and Fig. 7.

## 5 Computational experiments and comparison

In this section, we test our approach using realistic data from the first affiliated hospital of University of Science and Technology of China. The data instances are based on the available database of surgical procedures collected from 2017. Statistics in this year show that the hospital received up to 3,680,000 patients and performed 63,000 surgeries. The number of elective patients is 600 each week. The specialty of each patient is given and the surgery duration of each specialty follows a normal distribution. The surgical risk coefficient $w_i$ are set to be (0.2, 0.4, 0.6, 0.8, 1.0). The hospitalization day cost has been set to 366.41 euros (Guinet and Chaabane 2003). The overtime cost of an OR is set to 7.06 euros per minute (Jebali et al. 2006).The parameter settings are shown in Table 6:

We group all the surgery departments into three main specialties. The benefit of grouping is that when faced with emergencies, the head nurse can adjust operating time between surgery departments within a group without fear of the deviation from the optimized solution. Therefore, minor adjustments could be carried out without making completely new plans. The first specialty includes gynecology, ophthalmology and urology. The second specialty consists of general, oral, otolaryngology, and vascular surgery. The third specialty consists of neurosurgery, organ transplantation, orthopedics, surgical tumor and thoracic surgeries. The duration of each specialty of surgeries is random, following a known normal distribution with a mean μ and a

**Table 5** Procedure of GWO-VNS algorithm

| GWO-VNS | |
|---|---|
| Input: | $f()$, popsize, dimention (dim), max_iter. |
| Output: | An optimized solution and a best objective function value. |
| 1. | Initialize the grey wolf population |
| 2. | Initialize $a, A, C$ |
| 3. | Encoding correction |
| 4. | Execute Heuristic Algorithm 1&2 |
| 5. | Calculate the fitness of each search agent |
| 6. | Sort the search agent by fitness |
| 7. | Set $X_\alpha$, $X_\beta$, and $X_\delta$ as the first, second and third best solutions, $f(X_\alpha)$ as best fitness |
| 8. | Set $iter = 1$ |
| 9. | While $(iter < max\_iter)$, do |
| 10. | Execute VNS-based local search procedure for $X_\alpha$, $X_\beta$, and $X_\delta$ to obtain new solutions $X_\alpha^{new}$, $X_\beta^{new}$, and $X_\delta^{new}$ |
| 11. | Encoding correction |
| 12. | If $f(X_\alpha^{new}) < f(X_\alpha)$, then |
| 13. | $f(X_\alpha) \leftarrow f(X_\alpha^{new})$, $X_\alpha \leftarrow X_\alpha^{new}$ |
| 14. | End if |
| 15. | If $f(X_\beta^{new}) < f(X_\beta)$, then |
| 16. | $f(X_\beta) \leftarrow f(X_\beta^{new})$, $X_\beta \leftarrow X_\beta^{new}$ |
| 17. | End if |
| 18. | If $f(X_\alpha^{new}) < f(X_\delta)$, then |
| 19. | $f(X_\delta) \leftarrow f(X_\alpha^{new})$, $X_\delta \leftarrow X_\alpha^{new}$ |
| 20. | End if |
| 21. | For $i = X^1$: $X^{popsize}$ |
| 22. | For $j = 1$: dim |
| 23. | Update $a$, $A$ and, $C$, $r1$ and $r2$ |
| 24. | Calculate $\overrightarrow{D_\alpha}, \overrightarrow{D_\beta}, \overrightarrow{D_\delta}$ |
| 25. | Calculate $\overrightarrow{X_1}, \overrightarrow{X_2}, \overrightarrow{X_3}$ |
| 26. | Calculate $\vec{X}(t+1) = (\overrightarrow{X_1}, \overrightarrow{X_2}, \overrightarrow{X_3}) / 3$ |
| 27. | End for |
| 28. | Encoding correction |
| 29. | End for |
| 30. | Execute Heuristic Algorithm 1&2 |
| 31. | Calculate the fitness of each search agent |
| 32. | Sort the search agent by fitness |
| 33. | Update $X_\alpha$, $X_\beta$, $X_\delta$, and $f(X_\alpha)$ |
| 34. | $iter$++ |
| 35. | End while |
| 36. | Output $X_\alpha$ and $f(X_\alpha)$ |

**Table 6** Parameters setting

| Notation | Definition | Value |
|---|---|---|
| $I$ | The number of patients within the planning horizon | 400,500,600 |
| $S$ | The number of surgeons | 80,100 |
| $R$ | The number of ORs | 30,35,40 |
| $C$ | The number of specialties | 3 |
| $E$ | The max number of surgeons in each OR day | 8 |
| $C_{over}$ | Overtime cost per hour of each OR | 423.6 euros |
| $C_{hos}$ | Hospitalization cost per day | 366.41 euros |
| $\tau$ | $C_{over}/C_{hos}$ | 1.16 |

variance $\sigma^2$. For specialty 1, $\mu_1 = 1$ and $\sigma_1^2 = 0.5^2$. For specialty 2, $\mu_2 = 2$ and $\sigma_2^2 = 0.8^2$. For specialty 3, $\mu_3 = 4$ and $\sigma_3^2 = 1^2$.

In order to test the performance of our proposed algorithm GWO-VNS, a serial of computational experiments are conducted, compared with three classic algorithms: GWO (Mirjalili et al. 2014), VNS (Lei and Guo 2016), and PSO (Taherkhani and Safabakhsh 2016). According to the number of patients, surgeons and ORs, 18 instances are generated in our computational experiments. The average objective value (Ave) and the minimum objective value (Min) are measured over 18 instances in Table 7. We also analyze and compare the performance of these four methods by Relative Percent Deviation (RPD) (Vallada and Ruiz 2011), defined as follows:

$$RPD(M) = \frac{Ave(M) - BestF}{BestF} \times 100$$

where Ave(M) is the average value acquired by algorithm M. BestF denotes the best-known fitness, obtained by all four algorithms for solving the same case. Our goal is to acquire the minimum value, and thus the larger the RPD, the worse the performance. To ensure that the algorithms can converge to a good solution, we set the population size as 20, and the maximum number of iterations as 200. Each case is run for 20 times to ensure the reliability of experiments. The initial solution is the same for the four algorithms to ensure that they start at the same level to search for optimized solutions. We use a Lenovo computer running Windows 10 with an Intel(R) Core(TM)2 Duo CPU @2.93 GHz and 8 GB RAM to implement these four algorithms in Python. In Table 7, the last two rows show the best and average RPD (ARPD) values (the latter between brackets) of instances 1–9 and instances 10–18 for all methods. We can find that GWO-VNS has the best performance in obtaining average cost, minimum cost, best RPD and average RPD compared with other three algorithms.

Figures 8 and 9 provide us an intuitive means of data analysis. In Fig. 8, the cost increases significantly when the number of patients increases. Figure 9 shows that the RPD values of GWO-VNS and GWO are maximal when the number of surgeons, patients and ORs is 100, 600, 300. The RPD values of VNS and PSO are maximal when the number of surgeons, patients and ORs is 80, 500, 300. Figure 9 shows that the GWO-VNS has more stable RPD values than other three algorithms. The RPD

**Table 7** Computational results

| No. | S | I | R | GWO-VNS | | | GWO | | | VNS | | | PSO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Ave | Min | RPD | Ave | Min | RPD | Ave | Min | RPD | Ave | Min | RPD |
| 1 | 80 | 400 | 30 | 293,722.3 | 274,134.0 | 7.2 | 301,759.7 | 285,846.0 | 10.1 | 316,593.7 | 286,212.0 | 15.5 | 312,827.5 | 287,749.2 | 14.1 |
| 2 | 80 | 400 | 35 | 290,680.9 | 281,380.8 | 3.3 | 304,903.6 | 295,801.2 | 8.4 | 316,930.4 | 288,700.8 | 12.6 | 313,182.5 | 293,092.8 | 11.3 |
| 3 | 80 | 400 | 40 | 302,206.2 | 283,869.6 | 6.5 | 318,775.0 | 306,488.4 | 12.3 | 318,961.7 | 290,384.4 | 12.4 | 303,388.4 | 288,847.2 | 6.9 |
| 4 | 80 | 500 | 30 | 308,794.2 | 286,358.4 | 7.8 | 316,015.4 | 293,385.6 | 10.4 | 400,641.9 | 372,954.0 | 39.9 | 373,473.7 | 356,557.2 | 30.4 |
| 5 | 80 | 500 | 35 | 308,336.7 | 286,944.0 | 7.5 | 318,215.0 | 300,412.8 | 10.9 | 389,350.8 | 359,119.2 | 35.7 | 374,055.7 | 355,459.2 | 30.4 |
| 6 | 80 | 500 | 40 | 313,746.2 | 293,898.0 | 6.8 | 319,488.7 | 301,657.2 | 8.7 | 394,167.4 | 358,387.2 | 34.1 | 375,786.8 | 364,902.0 | 27.9 |
| 7 | 80 | 600 | 30 | 394,218.6 | 368,708.4 | 6.9 | 402,413.3 | 371,050.8 | 9.1 | 479,438.0 | 449,008.8 | 30.0 | 443,961.7 | 428,000.4 | 20.4 |
| 8 | 80 | 600 | 35 | 393,936.8 | 360,656.4 | 9.2 | 399,108.4 | 374,857.2 | 10.7 | 469,493.8 | 449,228.4 | 30.2 | 443,420.0 | 415,190.3 | 23.0 |
| 9 | 80 | 600 | 40 | 395,675.3 | 367,317.6 | 7.7 | 403,661.4 | 392,059.2 | 9.9 | 477,816.7 | 450,985.2 | 30.1 | 443,961.7 | 414,092.4 | 20.9 |
| 10 | 100 | 400 | 30 | 297,938.6 | 276,183.6 | 7.9 | 305,412.4 | 281,600.4 | 10.6 | 323,888.0 | 297,558.0 | 17.3 | 316,934.0 | 301,364.4 | 14.8 |
| 11 | 100 | 400 | 35 | 300,046.8 | 283,723.2 | 5.8 | 304,072.8 | 290,604.0 | 7.2 | 314,862.5 | 292,873.2 | 11.0 | 315,111.4 | 296,826.0 | 11.1 |
| 12 | 100 | 400 | 40 | 301,071.6 | 284,821.2 | 5.7 | 307,948.7 | 297,484.8 | 8.1 | 321,069.8 | 297,118.8 | 12.7 | 316,893.8 | 299,680.8 | 11.3 |
| 13 | 100 | 500 | 30 | 377,049.5 | 364,755.6 | 3.4 | 388,688.3 | 368,196.0 | 6.6 | 406,421.0 | 379,761.6 | 11.4 | 400,118.5 | 374,784.0 | 9.7 |
| 14 | 100 | 500 | 35 | 383,670.5 | 350,994.0 | 9.3 | 394,979.9 | 379,102.8 | 12.5 | 397,289.3 | 371,197.2 | 13.2 | 399,419.5 | 381,372.0 | 13.8 |
| 15 | 100 | 500 | 40 | 377,188.6 | 342,429.6 | 10.2 | 388,300.4 | 369,074.4 | 13.4 | 397,121.0 | 371,416.8 | 16.0 | 399,313.3 | 381,591.6 | 16.6 |
| 16 | 100 | 600 | 30 | 399,748.9 | 348,944.4 | 14.6 | 401,545.9 | 383,860.8 | 15.1 | 480,949.6 | 445,275.6 | 37.8 | 449,305.3 | 424,779.6 | 28.8 |
| 17 | 100 | 600 | 35 | 403,734.6 | 362,779.2 | 11.3 | 404,635.0 | 375,003.6 | 11.5 | 488,522.2 | 462,843.6 | 34.7 | 460,116.9 | 446,520.0 | 26.8 |
| 18 | 100 | 600 | 40 | 405,996.5 | 381,298.8 | 6.5 | 411,175.4 | 386,276.4 | 7.8 | 472,707.3 | 445,129.2 | 24.0 | 443,332.1 | 409,188.0 | 16.3 |
| (1–9) | ARPD | | | | 3.3(6.9) | | | 8.4 (10.0) | | | 12.4 (26.7) | | | 6.9 (20.5) | |
| (10–18) | ARPD | | | | 3.4(8.3) | | | 7.2 (10.3) | | | 11.0 (19.7) | | | 9.7 (16.5) | |

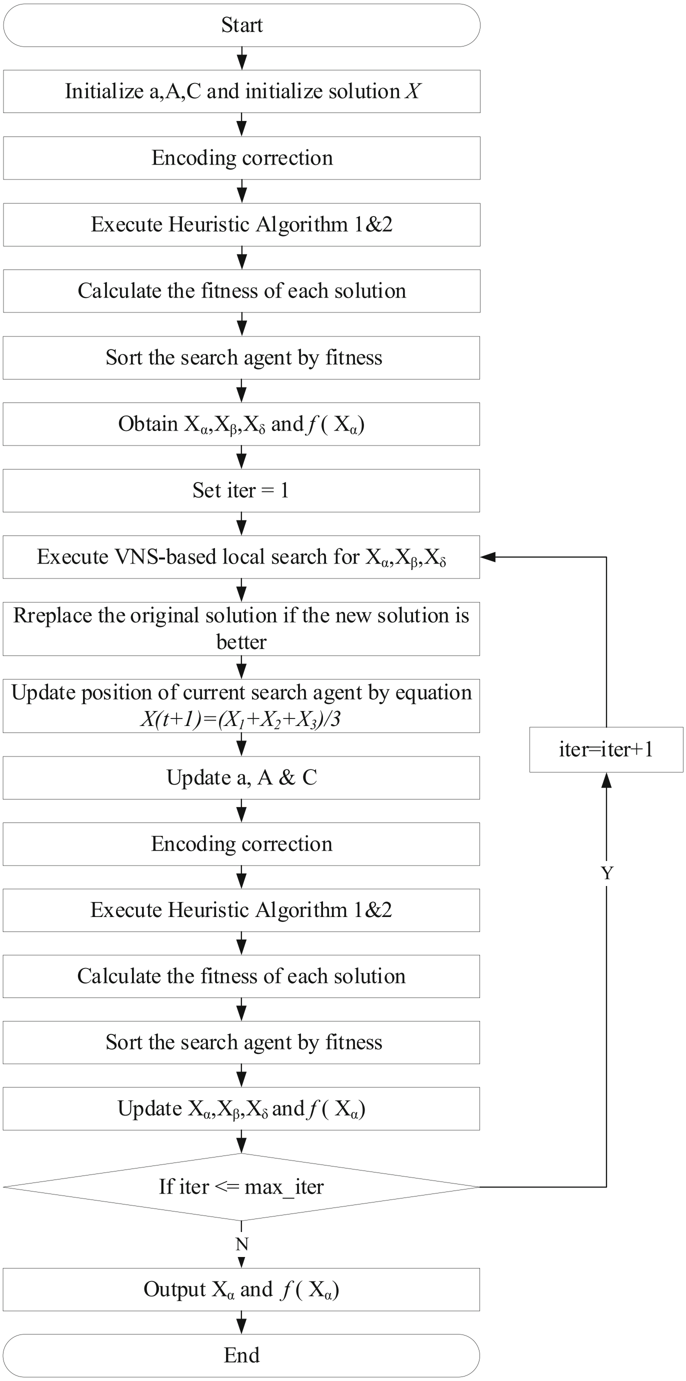



**Fig. 7** The flowchart of the hybrid GWO-VNS

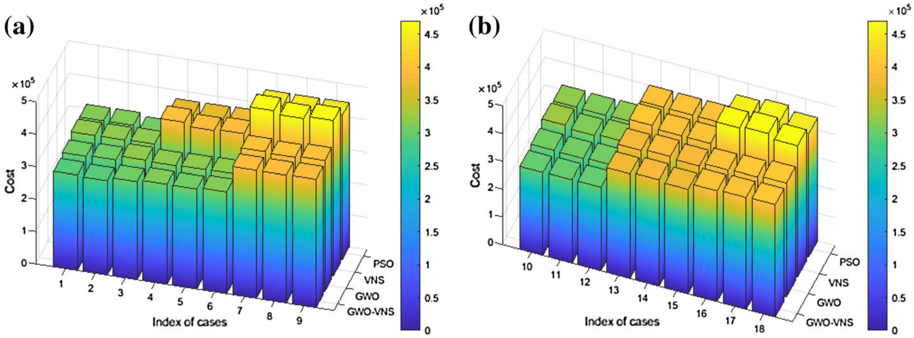



**Fig. 8** Cost in different numerical cases

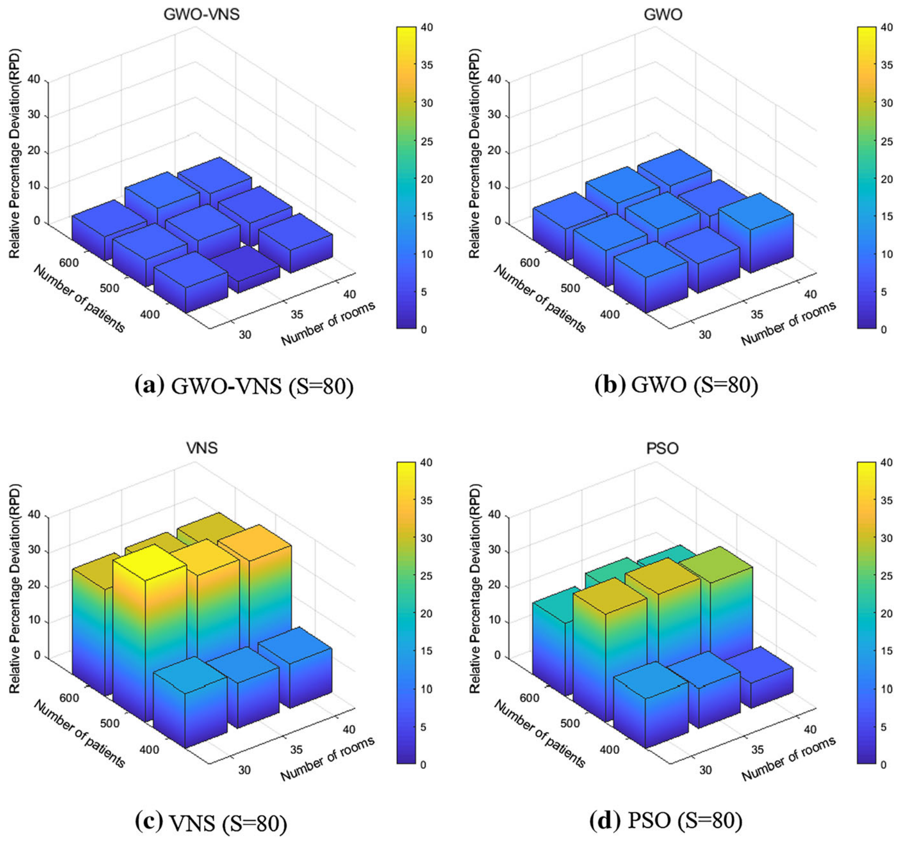
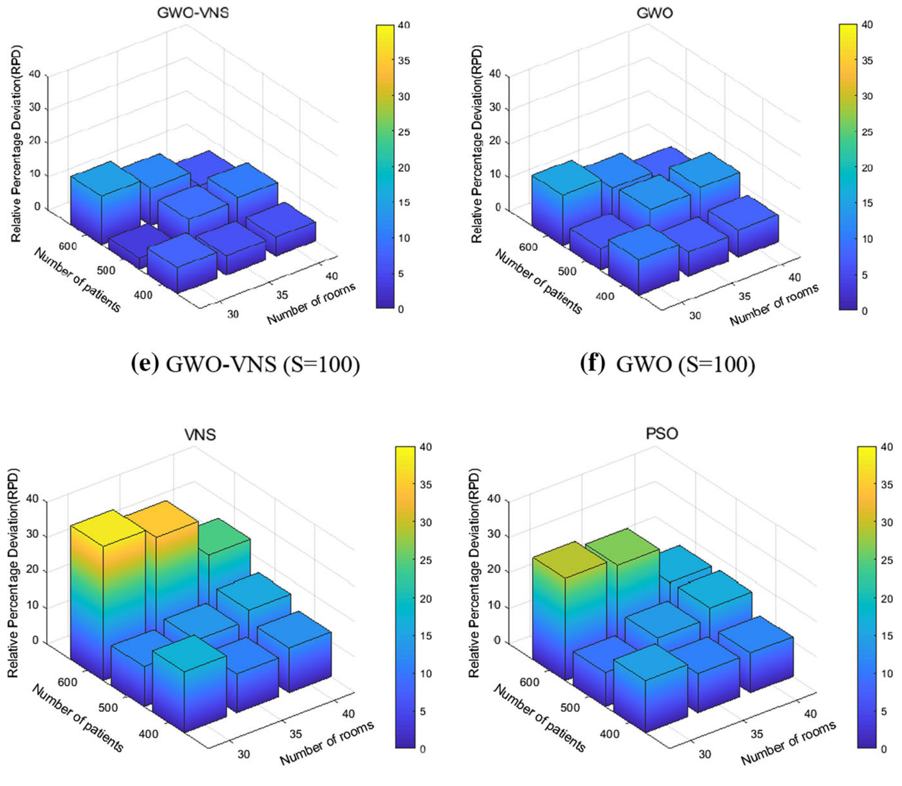



**Fig. 9** RPD results with different numbers of surgeons, patients and ORs




**(e)** GWO-VNS (S=100) **(f)** GWO (S=100)

**(g)** VNS (S=100) **(h)** PSO (S=100)

**Fig. 9** continued

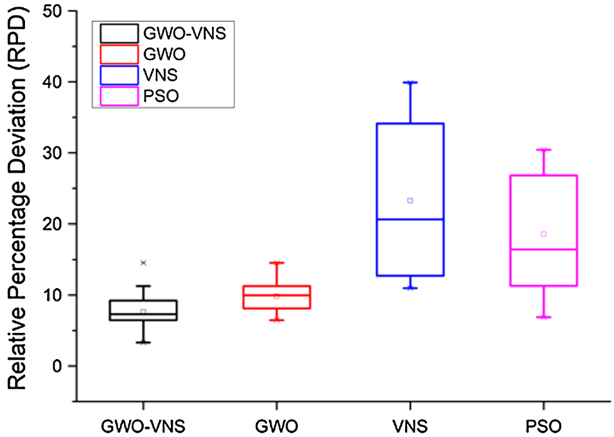



**Fig. 10** The box-plot of RPD

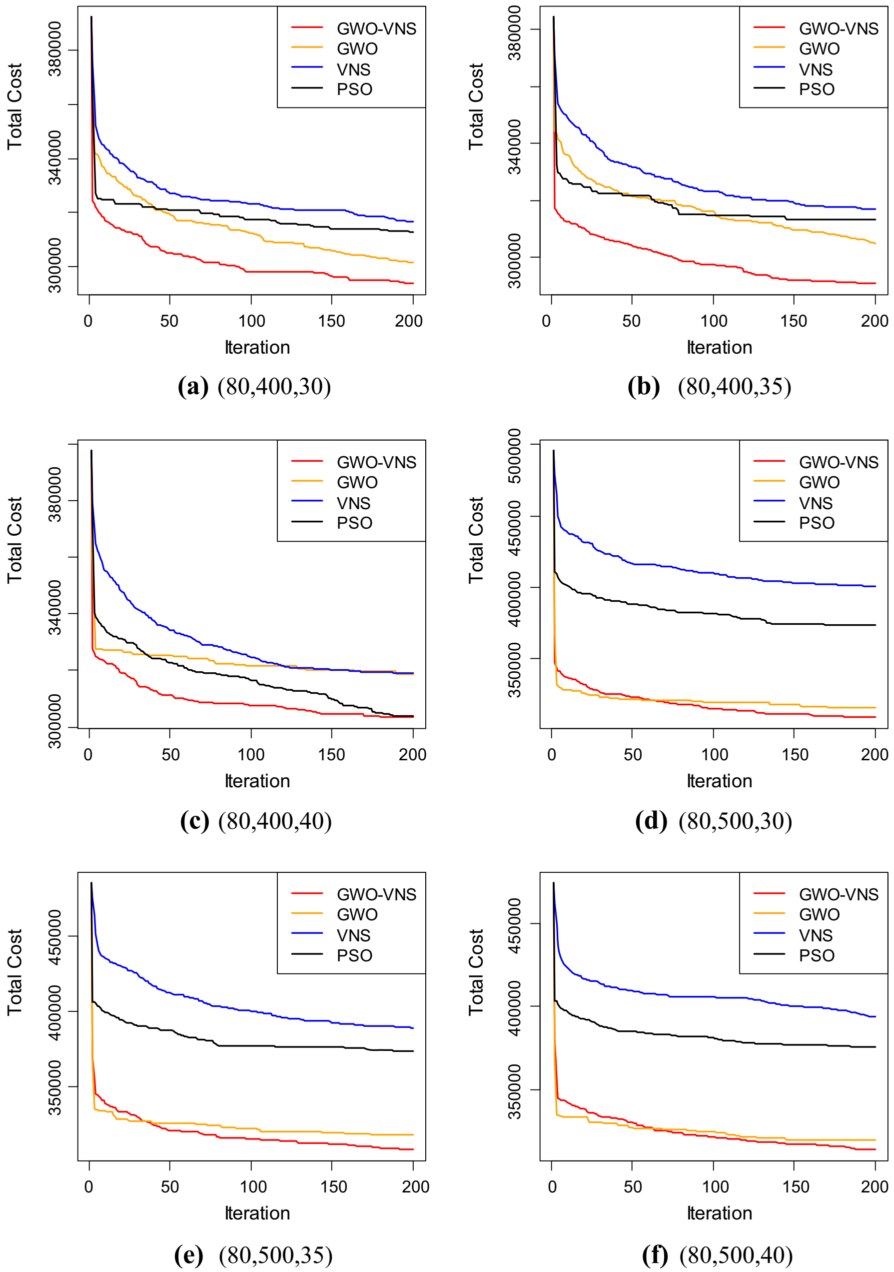
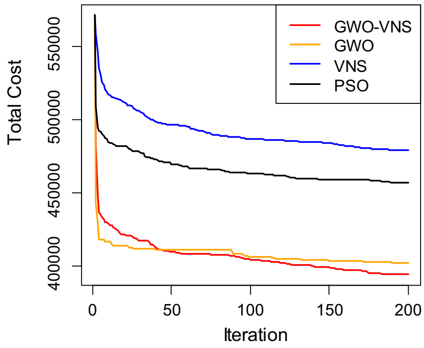
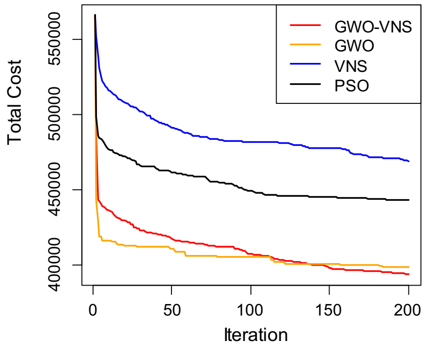


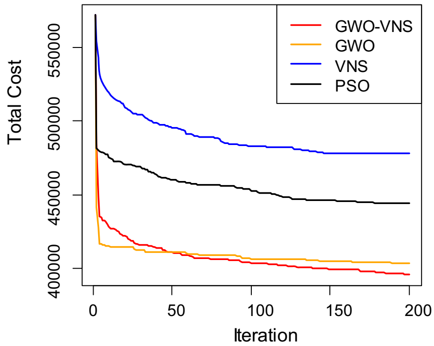

**Fig. 11** Convergence curves for 18 instances



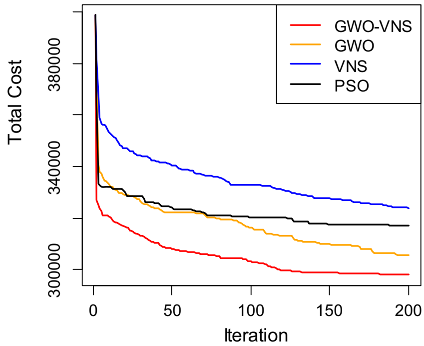

**(g)** (80,600,30)



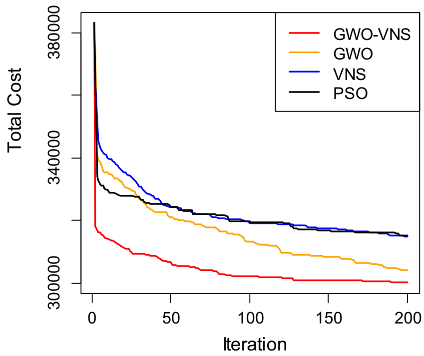

**(h)** (80,600,35)



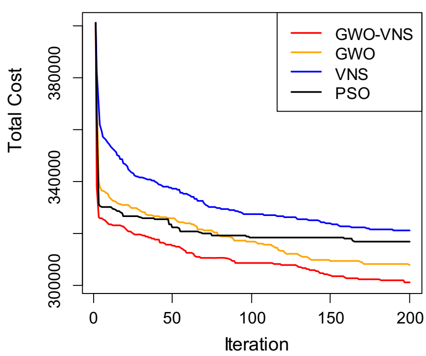

**(i)** (80,600,40)




**(j)** (100,400,30)

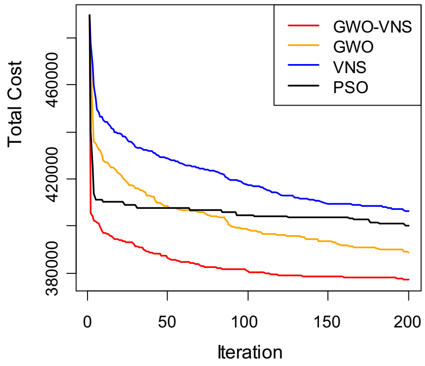



**(k)** (100,400,35)

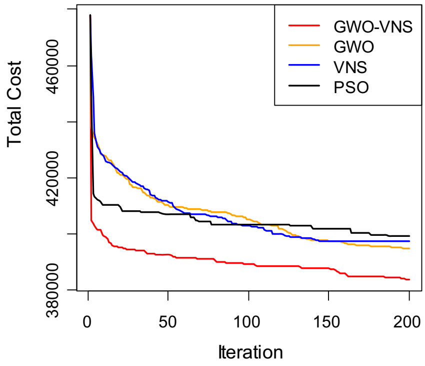



**(l)** (100,400,40)

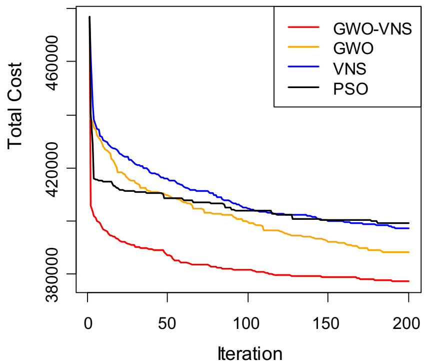
**Fig. 11** continued



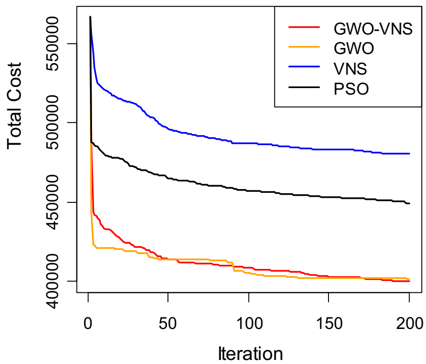

**(m)** (100,500,30)



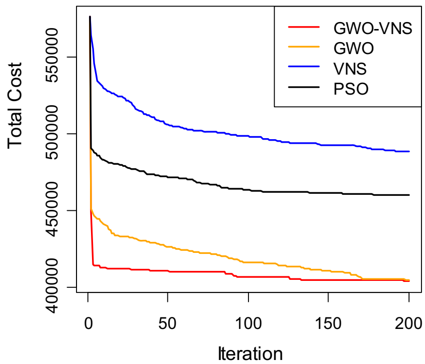

**(n)** (100,500,35)



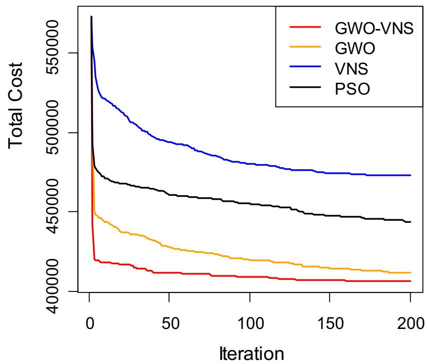

**(o)** (100,500,40)




**(p)** (100,600,30)




**(q)** (100,600,35)




**(r)** (100,600,40)

**Fig. 11** continued

values of GWO are smaller than those of VNS and PSO. The RPD values of VNS and PSO are similar. Particularly, the RPD values of GWO-VNS are very small when S = 80, I = 400. VNS and PSO are especially unstable and GWO cannot converge to get a best value. From Figs. 8 and 9 we can obtain the deduction that GWO-VNS is more stable and efficient compared with other three algorithms.

We can exactly see the differences of RPD values among the four algorithms from the box plot graphic in Fig. 10, where the minimum, the upper and lower quartiles, median, maximum and mean value for 18 cases are shown. We can find that the confidence intervals of VNS and PSO are overlapped. It demonstrates that they are not statistically different, and the performance of these two algorithms is at the same level. GWO obtains smaller minimum, upper and lower quartiles, median, maximum and mean value than those of VNS and PSO. Additionally, the upper and lower quartiles, median, mean value, and the difference between the upper and lower quartiles of GWO-VNS are much smaller than other three selected algorithms. This is a clear demonstration of the better performance of GWO and GWO-VNS than VNS and PSO, and the best performance of GWO-VNS among the all four algorithms. At the same time, this conclusion is consistent with the results in Table 7 and Fig. 8.

The convergence curve graphs of GWO-VNS, GWO, VNS, PSO for the eighteen instances are shown in Fig. 11. The average of best solution values in each iteration is listed in each figure. From Fig. 11, we can see that the differences of the best solutions among GWO-VNS, GWO, VNS, and PSO become smaller with the number of surgeons increasing or with the number of patients decreasing. GWO gets better solutions than VNS and PSO except for the instance (80,400,40). Compared with GWO, VNS, and PSO, GWO-VNS can always get better solutions in approaching solutions to these instances and its convergence speed is faster as well. Moreover, the convergence curves demonstrate the greater searching ability of our GWO-VNS with the number of patients increasing. Additionally, GWO, VNS, and PSO all converge to a local optimum after one hundred and fifty iterations. However, only GWO-VNS can overstep the local extremum and continue to search for better solutions. Based on above description and discussion, we come to the conclusion that our GWO-VNS is stable and effective in the respect of solution quality as well as convergence speed. The results show that the GWO-VNS algorithm not only outperforms other algorithms in solving quality and convergence speed, but also maintains robustness in all cases.

## 6 Conclusion

In this paper, we solve the multi-functional OR scheduling problem integrating three decision stages. The first stage is to assign different OR days to specialties, the second stage is to address the surgeon assignment problem, and the third stage is to deal with the patient assignment and sequence problem. The objective is to minimize the total sum of patient waiting costs and operating room overtime costs. We use a hybrid VNS-GWO algorithm which is combined of GWO and VNS to solve the studied problem, where two heuristic algorithms are incorporated to determine the assignment of patients. A set of experiments are conducted to test the performance of our algorithm, compared with GWO, VNS, and PSO. The experimental results show that our hybrid algorithm has better performance in terms of solution quality and convergence speed.

Our proposed method offers decision makers the option of weighing multiple conflicting goals from different perspectives and determining the most suitable schemes. The results provide promising insights into patient waiting list management and limited medical resource optimization, showing that hospitals could handle surgeries at lower cost by scheduling specialties, ORs, surgeons effectively and provide satisfactory service for the patients. To be specific, our approach can find the best number of ORs for each specialty and the best assignment for each surgeon in the experiment, and thus, can provide useful information for reducing operating costs from both sides of hospitals and patients.

In the future research, we could consider other multi-objective problems such as the minimum number of ORs and required beds, the maximum makespan, or the maximum number of surgical cases scheduled. Furthermore, we could focus more on the structural properties of the optimized patient allocation problem from the application perspective. Additionally, we could extend our model and methods under more realistic assumptions, for example, the number of OR days assigned to each surgeon is not fixed, the surgery durations are random, or the arrival times of patients are uncertain. Finally, we could test the algorithm on open-source test instances (e.g., VRP problem or job scheduling problem). By further research, we will develop more effective algorithms to solve the practical problems and offer more managerial insights for the healthcare practitioners.

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (Nos. 71601065, 71690235 and 71690230), the Key research and development projects in Anhui (1804b06020377), the Basic scientific research projects in central colleges and Universities (JZ2018HGTB0232), and Innovative Research Groups of the National Natural Science Foundation of China (71521001).

## References

Aringhieri R, Landa P, Soriano P et al (2015) A two level metaheuristic for the operating room scheduling and assignment problem. Comput Oper Res 54:21–34

Astaraky D, Patrick J (2015) A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling. Eur J Oper Res 245:309–319

Cardoen B, Demeulemeester E, Beliën J (2006) Optimizing a multiple objective surgical case scheduling problem. DTEW-KBI_0625, pp 1–38

Choi S, Wilhelm WE (2014) An approach to optimize block surgical schedules. Eur J Oper Res 235:138–148. https://doi.org/10.1016/j.ejor.2013.10.040

Denton BT, Miller AJ, Balasubramanian HJ, Huschka TR (2010) Optimal Allocation of Surgery Blocks to Operating Rooms Under Uncertainty. Oper Res 58:802–816

Guido R, Conforti D (2017) A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. Comput Oper Res 87:270–282

Guinet A, Chaabane S (2003) Operating theatre planning. Int J Prod Econ 85:69–81

Hansen P, Mladenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130:449–467

Holte M, Mannino C (2013) The implementor/adversary algorithm for the cyclic and robust scheduling problem in health-care. Eur J Oper Res 226:551–559

Hosseini N, Taaffe KM (2015) Allocating operating room block time using historical caseload variability. Health Care Manag Sci 18:419–430

Jebali A, Hadj Alouane AB, Ladet P (2006) Operating rooms scheduling. Int J Prod Econ 99:52–62
Lee S, Yih Y (2014) Reducing patient-flow delays in surgical suites through determining start-times of surgical cases. Eur J Oper Res 238:620–629
Lei D, Guo X (2016) Variable neighborhood search for the second type of two-sided assembly line balancing problem. Comput Oper Res 72:183–188
Lin Y-K, Chou Y-Y (2019) A hybrid genetic algorithm for operating room scheduling. Health Care Manag Sci
Marques I, Captivo ME, Vaz Pato M (2015) A bicriteria heuristic for an elective surgery scheduling problem. Health Care Manag Sci 18:251–266
Min D, Yih Y (2010) Scheduling elective surgery under uncertainty and downstream capacity constraints. Eur J Oper Res 206:642–652
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
Penn ML, Potts CN, Harper PR (2017) Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables. Eur J Oper Res 262:194–206
Pham DN, Klinkert A (2008) Surgical case scheduling as a generalized job shop scheduling problem. Eur J Oper Res 185:1011–1025
Qu X, Peng Y, Kong N, Shi J (2013) A two-phase approach to scheduling multi-category outpatient appointments–A case study of a women's clinic. Health Care Manag Sci 16:197–216
Riise A, Mannino C, Burke EK (2016) Modelling and solving generalised operational surgery scheduling problems. Comput Oper Res 66:1–11
Saadouli H, Jerbi B, Dammak A et al (2015) A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery. Comput Ind Eng 80:72–79
Taherkhani M, Safabakhsh R (2016) A novel stability-based adaptive inertia weight for particle swarm optimization. Appl Soft Comput J 38:281–295
Tànfani E, Testi A (2010) A pre-assignment heuristic algorithm for the Master Surgical Schedule Problem (MSSP). Ann Oper Res 178:105–119
Testi A, Tanfani E, Torre G (2007) A three-phase approach for operating theatre schedules. Health Care Manag Sci 10:163–172
Vallada E, Ruiz R (2011) A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. Eur J Oper Res 211:612–622
Van Houdenhoven M, Van Oostrum JM, Hans EW et al (2007) Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling. Anesth Analg 105:707–714
Vancroonenburg W, Smet P, Vanden Berghe G (2015) A two-phase heuristic approach to multi-day surgical case scheduling considering generalized resource constraints. Oper Res Heal Care 7:27–39
Vijayakumar B, Parikh PJ, Scott R et al (2013) A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. Eur J Oper Res 224:583–591

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