A Two-Stage Approach for Resource Allocation and Surgery Scheduling With Assistant Surgeons

WANG, Jin; LI, Xin; CHU, Jing; TSUI, Kwok-Leung

Published in:
IEEE Access

Published: 01/01/2020

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2020.2979519

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher’s copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.
A Two-Stage Approach for Resource Allocation and Surgery Scheduling With Assistant Surgeons

JIN WANG, XIN LI, JING CHU, AND KWOK-LEUNG TSUI

1School of Intelligent Systems Science and Engineering, Institute of Physical Internet, Jinan University at Zhuhai, Zhuhai 519000, China
2Research Center for Modeling and Optimization of Complex Management Systems, College of Management, Shenzhen University, Shenzhen 518060, China
3Department of Urology, Zhuhai People’s Hospital (Zhuhai Hospital affiliated with Jinan University), Zhuhai 519000, China
4School of Data Science, City University of Hong Kong, Hong Kong

Corresponding authors: Xin Li (xli@szu.edu.cn) and Jing Chu (chujing-001@163.com)

ABSTRACT The high costs of operating rooms (ORs) make surgery scheduling decisions critical for hospitals, which refer to resource allocations and determining the start time of each surgery. We take into account an important issue that could affect surgery scheduling decisions, ie., assistant surgeons. The operating room scheduling problem is considered as a two-stage problem. The first stage refers to resource allocations, including allocating surgeries into ORs and assigning assistant surgeons to surgeries, while the start time of each surgery is determined in the second stage. The robustness of the model is considered in a tractable way. We develop a bound-based algorithm to solve the model with the objective of minimising the total cost in the two stages. Numerical studies show the good performances of the proposed algorithm. Sensitivity analysis is conducted to examine the effects of the key parameters. An easy-to-implement policy for decisions in the first stage is inspired by the results of the experiments.

INDEX TERMS Assistant surgeon, mixed integer programming, surgery scheduling, two-stage approach.

I. INTRODUCTION

Due to the high costs of operating rooms (ORs), resource allocation and surgery scheduling decisions are critical for maintaining a hospital. Such decisions usually include allocating surgeries into ORs and determining the start time of surgeries. [1]–[3]. In decades, researchers focus on how to make use of the resources in ORs under the uncertainties in surgery durations. However, an important issue that could impact surgery scheduling decisions is neglected in the literature, i.e., assistant surgeons. In particular, a surgery often needs two surgeons, the main surgeon, and an assistant. The assistant surgeon can help avoid costly errors and fatigue-related accidents. Thereby a surgeon who works for a surgery as the main surgeon could be an assistant for another surgery.

Resource allocation decisions in this paper refer to allocating surgeries into different ORs and assigning assistant for surgeries, and surgery scheduling decisions refer to determining the surgery sequence and the start time. The allocation and the sequence decisions could be formulated as binary variables, while the start time should be continuous variables.

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

Hence, a mixed integer programming (MIP) model is often formulated to resolve the problem [1]. However, MIP models are hard to be solved for large scale problems. In this paper, we propose a two-stage approach to solve the surgery scheduling problem with assistant surgeons. In the first stage, we solve a resource allocation problem, determining the surgery allocation and the surgeon assignment, i.e., assigning an assistant surgeon for each surgery. In the second stage, with the allocation results in the first-stage model, we determine the start time of each surgery. The main task of the second-stage modelling is to formulate the linear constraints relevant to the waiting time of the surgeons, including both the main surgeons and the assistant surgeons. Additionally, we consider the model robustness in a tractable way. A bound-based algorithm is proposed to solve the two-stage model. Numerical studies show the performances of the proposed algorithm. Sensitivity analysis is conducted to show the effects of the key parameters. An easy-to-implement policy is inspired, which performance is tested.

The main contributions of this paper are as follows:

- We propose a two-stage model for surgery scheduling with consideration of assistant surgeons.
- We develop a bound-based algorithm to solve the two-stage model.
• An easy-to-implement policy for decisions in the first stage is proposed.

The remainder of this paper is organised as follows. Section II reviews the literature related to surgery scheduling, while Section III introduces the two-stage model. Section IV proposes a bound-based algorithm to solve the model. Section V describes the findings of the numerical experiment, while Section VI presents the concluding remarks and suggestions for future research.

II. LITERATURE REVIEW

Researches have been focused on surgery scheduling problems. The reviews can be found in [4]–[6]. Our review would pay more attention to the impacts of assistant surgeon on surgeries and surgery scheduling models.

In terms of the effects of assistant surgeons or surgical team, the empirical study in [7] concluded that for improving OR efficiency it was important to decrease surgical team size and enhance communication between team members. Obviously, with the consideration of assistant surgeons, surgery scheduling models would be reformulated. However, few papers took it into consideration. Molina-Pariente et al. [8] solved an integrated operating room planning and scheduling problem with the influence of the surgical team size (one or two surgeons) as well as the assistant surgeon’s experience.

Plenty of studies focused on scheduling in healthcare, which greatly impacted the efficiency of healthcare systems. Zhang et al. [9] formulated the healthcare service configuration as a resource constrained project scheduling problem. Most of the decisions in literature relevant to surgery scheduling referred to assigning surgeries into different ORs, sequencing surgeries, as well as determining the start time of each surgery [10]–[12]. The sequence of surgeries and the start time of each surgery were also considered in [1], [13], [14]. Some papers took into account the selection of patients on a waiting list. Wang et al. [15] developed an integrated approach to select surgeries from the waiting list to perform in a day and determine the start time for each surgery. In this paper, we consider a surgery scheduling problem with assigning resources, sequencing surgeries and determining the start time.

Robust and stochastic models were proposed to solve problems with uncertainties in different areas [16], e.g., production control [17]–[19], transportation scheduling [20], as well as healthcare management [21]. Particularly, two-stage optimisation models were very useful for scheduling problems, including two-stage robust models [22]–[25] and two-stage stochastic models [26]. In healthcare, two-stage optimisation models were often proposed to address the uncertainty of surgery durations [24], [25]. In the first stage, allocation problem was considered, which was formulated as an integer programming problem, while the start time was considered in the second stage. Neyshabouri and Berg [24] proposed a two-stage robust optimisation model for scheduling elective surgeries as well as planning the downstream capacity. The uncertainties of both surgery duration and the length of stay (LOS) after the surgery were considered. Min and Yih [25] proposed a two-stage stochastic programming model to optimise the elective surgery planning problem. Similarly, uncertainty in both surgery duration and the LOS in the SICU was taken into account. In these papers, a mixed integer programming (MIP) model was proposed first, which then was transformed into a two-stage robust model. The first-stage model was an integer programming (IP) model, while the second-stage model was a linear programming (LP) model with continuous variables. However, in this paper, we propose the models in the two stages separately, which are solved consecutively. Different from literature we determine the sequence and start time in the second stage, which is formulated as an MIP model.

III. MODEL FORMULATION

This section presents the two-stage approach for surgery scheduling with assistant surgeons. In the first stage optimisation problem, the subject of focus is the resource allocation problem, including allocating surgeries into ORs and assigning an assistant surgeon for each surgery, while the second stage determines the start time for each surgery.

A. SETTINGS

There are some settings that should be claimed in advance. First, like [27], emergency surgeries are not considered, since they are performed in dedicated rooms. Second, the main surgeon of each surgeries is given. Surgeon assignment in this paper only refers to assigning an assistant surgeon for each surgery. Overall, the modelling ideas are the following: (1) in the first-stage optimisation problem, we assign an OR and an assistant surgeon for each surgery; and (2) based on the solutions of the first-stage model, the start time for each surgery is determined. Additionally, there is a constraint relevant to OR allocation, which is widely used in practice. That is, the surgeries of a surgeon should be assigned into the same OR. Through this arrangement, unnecessary switch time can be avoided, and waiting time (due to the overtime of another surgery) can also be reduced.

B. THE FIRST-STAGE OPTIMISATION MODEL: RESOURCE ALLOCATION

The surgeons are divided into three levels: \( L_1 \), \( L_2 \), and \( L_3 \). For notation convenience, \( L_i \) \((i = 1, 2, 3)\) also denotes the set of \( L_i \) surgeons as well as the number of \( L_i \) surgeons. As stated earlier, a surgery generally requires two surgeons: one as the so-called main surgeon and the other as the assistant. Note that an \( L_1 \) surgeon is a high-level surgeon who can only work as a main surgeon, an \( L_2 \) surgeon may be a main surgeon or an assistant, and an \( L_3 \) surgeon can only work as an assistant. Everyday surgeries by \( L_1 \) and \( L_2 \) surgeons should be allocated to several ORs. In reality, different surgeons may conduct different numbers of surgeries per day. Suppose that there are \( I \) surgeries in a day. Let \( i \) be the \( i \)th \((1 \leq i \leq I)\) surgery. For notation convenience, \( I \) is also used to represent the set containing all surgeries. \( I_m \) is defined as a set of the surgeries by Surgeon \( m \). Specially, \( I_{L_1}(I_{L_2}) \) denotes all surgeries by
L1(L2) surgeons. R is used as the number of ORs as well as the set of ORs. cA,m represents the cost if surgeon m is assigned as an assistant per minute. The values are different for different surgeons. However, the values for L2 surgeons are larger than those for L3 surgeons. The notations used in the first-stage model are summarized in Table 1. The first-stage optimisation model is as follows:

\[
\begin{align*}
\min & \quad \sum_{r \in R} c^R e_r + \sum_{m \in L_2 \cup L_3} \sum_{i \in I} d_i c^A_m b_{im} \\
\text{subject to} & \quad \sum_{m \in L_2 \cup L_3} b_{im} = 1 \quad \forall i \in I \quad (1) \\
& \quad b_{im} = 0 \quad \forall i \in I_m, \forall m \in M \quad (2) \\
& \quad h_{ir} = h_{ir} \quad \forall r \in R; i, i' \in I_m, m \in M \quad (3) \\
& \quad h_{ir} \leq e_r \quad \forall r \in R, i \in I \quad (4) \\
& \quad e_r \geq e_{r+1} \quad 1 \leq r \leq R - 1 \quad (5) \\
& \quad e_r \leq \sum_{i \in I} h_{ir} \quad \forall r \in R \quad (6) \\
& \quad b_{im}, h_{ir}, e_r \in [0, 1] \quad \forall i \in I, r \in R, m \in M. \quad (7)
\end{align*}
\]

In the model, the objective function includes the costs relevant to the decision variables in the first stage, including the OR opening costs and the costs of surgeons working as assistants. Constraint (2) ensures that exactly one surgeon is assigned as an assistant, and the assistant surgeon cannot be L1 surgeons. Constraint (3) indicates that a surgeon cannot work as an assistant surgeon of his/her own surgeries. Constraint (4) clarifies that a surgery can only be allocated to exactly one OR, while Constraint (5) requires that all of the surgeries of one surgeon must be assigned to the same OR. Constraint (6) shows that an OR must be open if a surgery is allocated to it, while Constraint (7) requires that ORs are opened sequentially, which breaks symmetry. Constraint (8) requires an OR should be closed if there is no surgery allocated in it. Constraint (9) is the binary constraint for the variables.

**C. THE SECOND-STAGE OPTIMISATION MODEL:**

**APPOINTMENT SCHEDULING**

In the second stage, the start time for each surgery would be determined based on the solutions in the first stage. Some information regarding the allocations can be obtained according to the first-stage model solutions. First, the set of the surgeries in which Surgeon m works as an assistant is defined as Q^A_m = \{i | b_{im} = 1\} \quad \forall m \in M,

while the set that includes the surgeries in which Surgeon m works as a main surgeon is defined as I_m. By combining the two sets, we obtain a new set denoted as Q_m containing all surgeries that Surgeon m (m ∈ M) works for (i.e., the surgeries where a surgeon works as either a main surgeon or an assistant).

\[
Q_m = Q^A_m \cup I_m \quad \forall m \in M
\]

Let Q = \{Q_m, m \in M\}. Similarly, we define a set H_r (r ∈ R) as a set including the surgeries that are allocated in OR r. That is,

\[
H_r = \{i | h_{ir} = 1\} \quad \forall r \in R
\]

Let H = \{H_r, r \in R\}.

The notations in the second-stage model are listed in Table 2. In particular, m(i), a(i), and r(i) are denoted as the main surgeon of surgery i, the assistant surgeon of Surgery i,

<table>
<thead>
<tr>
<th>Notation</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>c^W_m</td>
<td>parameter, the cost per unit waiting time for surgeon m, m ∈ M</td>
</tr>
<tr>
<td>t^0</td>
<td>parameter, the overtime cost of an OR</td>
</tr>
<tr>
<td>w^A_i</td>
<td>performance indicator, the waiting time of the surgeon</td>
</tr>
<tr>
<td>t^A_i</td>
<td>performance indicator, the waiting time of the assistant of surgery i</td>
</tr>
<tr>
<td>h^M_r</td>
<td>performance indicator, the overtime of OR r</td>
</tr>
<tr>
<td>t^i</td>
<td>performance indicator, the available time of the main surgeon for surgery i</td>
</tr>
<tr>
<td>t^R_i</td>
<td>performance indicator, the available time of the assistant of surgery i</td>
</tr>
<tr>
<td>f^A_i</td>
<td>performance indicator, 1 if surgery i is the first surgery that its main surgeon performs</td>
</tr>
<tr>
<td>f^R_i</td>
<td>performance indicator, 1 if surgery i is the first surgery that its assistant surgeon performs</td>
</tr>
<tr>
<td>α_{ij}</td>
<td>a binary variable, 1 if Surgery i precedes Surgery j; 0 if Surgery j precedes Surgery i or there is no sequence constraint between the two surgeries</td>
</tr>
<tr>
<td>T^M_r, T^A_i, T^R_i</td>
<td>auxiliary variables</td>
</tr>
<tr>
<td>ζ_{ij}, η_{ij}, β_{ij}</td>
<td>binary auxiliary variables</td>
</tr>
<tr>
<td>(m(i))</td>
<td>a function indicating the main surgeon of surgery i</td>
</tr>
<tr>
<td>(a(i))</td>
<td>a function indicating the assistant surgeon of surgery i</td>
</tr>
<tr>
<td>(r(i))</td>
<td>a function indicating the OR where surgery i is allocated</td>
</tr>
<tr>
<td>t_i</td>
<td>decision variable, the start time of surgery i</td>
</tr>
</tbody>
</table>
and the OR where Surgery \( i \) is allocated, respectively. The model in the second stage is as follows:

\( \mathcal{H}(h, b) \)

\[
\min_{i \in I} \Bigg( c_{m(i)}^w w_i^M + c_{a(i)}^w w_i^A \Bigg) + \sum_r c^O O_r
\]

subject to \( t_i^M = \psi(Q) \quad \forall i \in I \)

\( t_i^A = \phi(Q) \quad \forall i \in I \)

\( t_i^R = \psi(H) \quad \forall i \in I \)

\( t_i = \max\{t_i^M, t_i^A, t_i^R\} \quad \forall i \in I \)

\( w_i^M = t_i - t_i^M \quad \forall i \in I \)

\( w_i^A = t_i - t_i^A \quad \forall i \in I \)

\( O_r = \left( \max (t_i + d_i) - T \right)^+ \quad \forall r \in R \)

\( t_i^M, t_i^A, t_i^R, t_i, w_i^M, w_i^A, O_r \geq 0 \quad \forall i \in I, r \in R \)

The objective function, \( \mathcal{H}(h, b) \), includes the waiting time costs of the main surgeon and the assistant, as well as the overtime costs of the ORs. Next, the constraints are explained in sequential order. Constraints (11-13) are about the available time of the main surgeon, the assistant and the OR, which are related to the allocation results \( \text{i.e., } H \text{ and } Q \) obtained by the solutions of the first-stage model. Constraint (14) requires that a surgery starts only if the main surgeon, the assistant, and the OR where the surgery is allocated are available. Constraints (15-17) are about the waiting time of the main surgeon and the assistant, as well as the overtime of the ORs. Since it is a minimisation model, Constraints (14) and (17) can be easily transformed into linear ones, which are presented in the following proposition:

**Proposition 1:** Constraint (14) is equivalent to

\[
t_i \geq t_i^M, \quad t_i \geq t_i^A, \quad t_i \geq t_i^R \quad \forall i \in I
\]

Constraint (17) is equivalent to

\[
O_r \geq Y_r - T \quad \forall r \in R
\]

\[
Y_r \geq t_i + d_i \quad \forall i \in H_r, \quad \forall r \in R
\]

\[
O_r \geq 0 \quad \forall r \in R
\]

**D. THE AVAILABLE TIME**

Based on the solutions of the first-stage model, we can obtain the information about whether two surgeries are allocated in the same OR, which can be described by a binary variable \( \beta_{ij} \) (\( i, j \in I \)). There is

\[
\beta_{ij} = \begin{cases} 1 & \text{if } h_{ir} = h_{jr} = 1, \quad \exists r \in R \\ 0 & \text{otherwise} \end{cases}
\]

Similarly, we use a binary variable \( \gamma_{ij} \) (\( i, j \in I \)) to denote whether two surgeries are performed by the same surgeon (who could be the main surgeon or the assistant surgeon). There is

\[
\gamma_{ij} = \begin{cases} 1 & \text{if } i \in Q_m, \quad j \in Q_m, \exists m \in M \\ 0 & \text{otherwise} \end{cases}
\]

To determine the available time of the surgeons, we define the surgery sequence first. \( \alpha_{ij} \) is defined as a binary variable, which equals 1 if Surgery \( i \) precedes Surgery \( j \), 0 if Surgery \( j \) precedes Surgery \( i \) or there is no sequence constraint between the two surgeries. The sequence has to satisfy Constraint (23-28). Constraint (23) ensures that Surgery \( j \) is behind Surgery \( i \) if Surgery \( i \) precedes Surgery \( j \). Constraint (24) requires that Surgery \( i \) precedes Surgery \( j \) if Surgery \( i \) precedes Surgery \( k \) and Surgery \( k \) precedes Surgery \( j \). Constraints (25-27) are about the fact that there is a sequence constraint only if the two surgeries are allocated in the same OR or they are performed by the same surgeon. Constraint (28) requires that surgeries of high-level surgeons precede those of lower-level surgeons.

\[
\alpha_{ij} + \alpha_{ji} \leq 1 \quad \forall i, j \in I, \quad i \neq j
\]

\[
\alpha_{ij} + \alpha_{ik} + \alpha_{jk} - 1 \quad \forall i, j, k \in I, \quad i \neq j \neq k
\]

\[
\alpha_{ij} + \alpha_{ji} \geq \beta_{ij} \quad \forall i, j \in I, \quad i \neq j
\]

\[
\alpha_{ij} + \alpha_{ji} \geq \gamma_{ij} \quad \forall i, j \in I, \quad i \neq j
\]

\[
\alpha_{ij} + \alpha_{ji} \leq \beta_{ij} + \gamma_{ij} \quad \forall i, j \in I, \quad i \neq j
\]

\[
\alpha_{ij} \geq \beta_{ij} \quad \forall i \in L_1, \quad j \in L_2
\]

We next formulate the available time of the main surgeon, the assistant surgeon and the OR for each surgeon. For the available time of the main surgeon, it needs to be considered that whether or not the surgery is the first one that the main surgeon performs in the day. A binary variable \( f_i^M \) is defined, which equals 0 if Surgery \( i \) is the first surgery that its main surgeon performs in the day, 1 otherwise. Hence,

\[
f_i^M \geq \alpha_{ij} \quad \forall j \in Q_{m(i)}, \quad \forall i \in I
\]

\[
f_i^M \leq \sum_{j \in Q_{m(i)}} \alpha_{ij} \quad \forall i \in I
\]

So Equation (11) can be expressed as

\[
t_i^M = \left\{ \begin{array}{ll} t_i & \text{if } f_i^M = 0 \\ \max_{j \in Q_{m(i)} \setminus i} (t_j + d_j) & \text{if } f_i^M = 1 \end{array} \right.
\]

That is, the available time equals the exact start time if the surgery is the first one the main surgery performs in the day; otherwise, it equal the ending time of the previous surgery that the main surgeon performs before the surgery (Surgery \( i \)). Since \( t_i^M \) is not directly related to the objective function like \( O_r \) in Equation (17), it cannot be simply handled using the inequalities like (20). Instead, \( t_i^M \) must exactly equal the maximum ending time of the surgeries that precede Surgery \( i \) if it is not the first surgery performed by the surgeon in the day. The linear formulations are as follows.

\[
T_{ji}^M \geq t_i + d_i - M(1 - \alpha_{ij}) \quad \forall i \in Q_{m(i)}, \quad \forall i \in I
\]

\[
T_{ji}^M \leq M\alpha_{ij} \quad \forall j \in Q_{m(i)}, \quad \forall i \in I
\]

\[
T_{ji}^M \leq t_j + d_j + M(1 - \alpha_{ij}) \quad \forall j \in Q_{m(i)} \setminus i, \quad \forall i \in I
\]

\[
T_{ji}^M \geq 0 \quad \forall j \in Q_{m(i)} \setminus i, \quad \forall i \in I
\]

\[
t_i^M \geq t_i - Mf_i^M \quad \forall i \in I
\]
\[ \begin{align*}
  i^M_j & \geq T^M_j - M(1 - f^M_j) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  i^A_i & \leq T^A_i + M(1 - \eta_j) + M(1 - f^A_i) & \forall j & \in Q_{a(i)}, \forall i \in I \\
  f^A_j & \geq \alpha_j & \forall j & \in Q_{a(i)}, \forall i \in I \\
  f^R_i & \leq \sum_{j \in Q_{a(i)}} \alpha_j & \forall i & \in I \\
  T^A_j & \leq T^A_i + M(1 - \alpha_j) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  T^R_i & \leq \sum_{j \in Q_{a(i)}} \alpha_j & \forall i & \in I \\
  T^A_j & \geq T^R_i + M(1 - \theta_j) + M(1 - f^A_i) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  T^R_i & \leq T^A_i + M(1 - \alpha_j) & \forall j & \in Q_{a(i)}, \forall i \in I \\
  i^A_i & \leq T^R_i - M(1 - f^R_i) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  \sum_{j \in Q_{a(i)}} \eta_j & = 1 & \forall i & \in I \\
 \end{align*} \]

where \( M \) is a large positive number, and \( \zeta_j \) is a binary variable. \( T^M_j \) represents the ending time of Surgery \( j \) if it precedes Surgery \( i \), 0 otherwise.

Similarly, the available time of the assistant surgeon can be formulated as follows.

\[ \begin{align*}
  t^A_i & \leq T^A_i + M(1 - \eta_j) + M(1 - f^A_i) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  f^A_j & \geq \alpha_j & \forall j & \in Q_{a(i)}, \forall i \in I \\
  f^R_i & \leq \sum_{j \in Q_{a(i)}} \alpha_j & \forall i & \in I \\
  T^A_j & \geq T^R_i + M(1 - \alpha_j) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  T^R_i & \leq T^A_i + M(1 - \alpha_j) & \forall j & \in Q_{a(i)}, \forall i \in I \\
  t^A_i & \leq T^R_i - M(1 - f^R_i) & \forall j & \in Q_{a(i)} \setminus i, \forall i \in I \\
  \sum_{j \in Q_{a(i)}} \eta_j & = 1 & \forall i & \in I \\
 \end{align*} \]

And the available time of the OR for each surgery can be formulated as follows.

\[ \begin{align*}
  t^R_i & \leq T^R_i + M(1 - \theta_j) + M(1 - f^R_i) & \forall j & \in H_{r(i)}, \forall i \in I \\
  f^R_i & \geq \theta_i & \forall j & \in H_{r(i)}, \forall i \in I \\
  f^A_i & \leq \sum_{j \in H_{r(i)}} \alpha_j & \forall i & \in I \\
  T^R_i & \leq T^A_i + M(1 - \alpha_j) & \forall j & \in H_{r(i)} \setminus i, \forall i \in I \\
  T^A_i & \leq T^R_i + M(1 - \alpha_j) & \forall j & \in H_{r(i)}, \forall i \in I \\
  t^A_i & \leq T^R_i - M(1 - f^A_i) & \forall j & \in H_{r(i)} \setminus i, \forall i \in I \\
  \sum_{j \in H_{r(i)}} \theta_j & = 1 & \forall i & \in I \\
 \end{align*} \]

\( T^A_j \) and \( T^R_i \) are similar with \( T^M_j \), while \( \eta_j \) and \( \theta_j \) are binary variables and similar with \( \zeta_j \). We summarize the linearized the second-stage model formally as the following proposition.

**Proposition 2:** The second-stage model is equivalent to

\[ \mathcal{H}(h, b) = \min \sum_{i \in I} \left( c^w_{a(i)} w^M_i + c^w_{a(i)} w^A_i \right) + \sum_r c^D_r O_r \]

s.t. (15), (16), (18) – (58)

### E. ROBUSTNESS

Surgery durations are in uncertain sets, i.e., \( d_i \in D_i \), where \( D_i \) is an uncertain set for surgery \( i \). In the field of robust optimization, the uncertain sets are assumed to have problems tractable. The widely used sets include ellipsoidal, polyhedral, cardinality constrained, and norm uncertainties [28]. In order to make the present model tractable, the following simple \( l_1 \)-norm uncertain set is used:

\[ D_i = \{ d_i : |d_i - \bar{d}_i| \leq \delta \bar{d}_i \} \]

where \( \bar{d}_i \) and \( \hat{d}_i \) are the mean and standard deviation of the surgery duration, respectively, and \( \delta \) is the parameter used to handle the uncertain set. Hence, the following is obtained:

\[ \bar{d}_i - \delta \bar{d}_i \leq d_i \leq \bar{d}_i + \delta \bar{d}_i. \]

To ensure that the constraints relevant to \( d_i \) (i.e., Constraint (21), (31), (33), (42), (44), (52) and (54)) are satisfied under the worst case of \( d_i \), the term, \( d_i \), in the constraints is substituted by \( \bar{d}_i + \delta \bar{d}_i \). In Section V, we will use numerical studies to test the policy performance under different settings of the value \( \delta \).

In sum, in this section a two-stage model was proposed to deal with the resource allocation and surgery scheduling problems in ORs. More specifically, in the first-stage, an IP model was developed to solve the surgery allocation and surgeon assignment problems. Based on the results of the first-stage model, the second stage proposed a linearized model for surgery scheduling problems. At last, a simple method was applied to deal with the robustness problem in the model.

### IV. ALGORITHM

This section proposes an algorithm that can be used to solve the two-stage model and a method to remove the symmetric solutions.

#### A. A BOUND-BASED ALGORITHM

It is difficult to combine the two models into one since the variables in the first-stage model are implicitly incorporated into the second-stage model. Hence, the first-stage model must be solved before the second-stage model. It is important to note that the first-stage model is an IP model, while the second-stage model is an MIP model. Due to the moderate scale problem (e.g., the total number of surgeries was less than 30), the two models can be efficiently solved. In fact, the scale is large enough to resemble actual cases. A simple method is to solve the first-stage model without \( \mathcal{H}(h, b) \) to obtain the solutions, after which the second-stage model can be solved. However, the solutions of this method are not optimal. Moreover, it is possible to obtain all the feasible solutions of the first-stage model, and then solve the second-stage model. However, it is computationally expensive, since solving an MIP model is relatively inefficient. Thus, a bound-based algorithm is proposed in order to avoid expensive computations. The details are in Table 3. The following function, i.e., the first-stage objective function without
Then, the relaxed second-stage model is solved to obtain the
objective value \( H_{y} \). Let \( Z_{UB} = Y_{bd} + H_{y} \).
Update \( Y_{bd} = Y_{bd} + \Delta \), where \( \Delta \) is a small number.

**Step 2:** Add the following constraint into the first-stage model:

\[
f(e, b) \geq Y_{bd}
\]

Solve the first-stage model without \( H(h, b) \) to obtain all the
optimal solutions, which is denoted as \( s_{j} \) \((s_{jn} \in S_{j}, 1 \leq n \leq N, \text{i.e., } N \text{ is the number of optimal solutions})\). Note that \( s_{jn} \) includes the values of \( h \) and \( b \). It is also possible to obtain the
objective value, which is denoted as \( U^{j} \). If the first-stage
model is unsolvable, then stop. Set \( n = 1 \).

**Step 3:** Using the results of the first-stage model, \( s_{jn} \), solve the relaxed
second-stage model to obtain the objective value \( H_{jn}^{R} \).

**Step 4:** If \( U^{j} + H_{jn}^{R} > Z_{UB} \), then let \( n = n + 1 \), and go to Step 3.

**Step 5:** Using the results of the first-stage model, \( s_{jn} \), solve the second-stage model to obtain optimal objective value \( H_{jn}^{R} \). If \( U^{j} + H_{jn}^{R} < Z_{UB} \), then record the solutions, \( h \) and \( b \), and update \( Z_{UB} = U^{j} + H_{jn}^{R} \). If \( n < N \), then let \( n = n + 1 \), and go to
Step 3.

**Step 6:** Let \( j = j + 1 \). If \( j > J \), then stop. Update \( Y_{bd} = Y_{bd} + \Delta \). If \( Y_{bd} > Z_{UB} \), then stop. Otherwise, go to Step 2.

\[ H(h, b) \text{ is defined as follows:} \]

\[
f(e, b) = \sum_{r \in R} \bar{c}_{r} + \sum_{i \in \ell, m} \sum_{l_{2} \in L_{3}} d_{i\ell m}^{A} h_{lm}
\]

In Step 1, the results of the first-stage model are used to solve the second-stage model and to obtain the corresponding objective values of \( Y_{bd} \) and \( H_{jn}^{R} \). In this case, \( Y_{bd} \) is considered as the lower bound of the first-stage model. In Step 2, the new first-stage model is solved to obtain a new set of the first-stage solutions and the corresponding objective value of \( U^{j} \). In Steps 3 to 5, each first-stage solution in Step 2 (denoted as \( s_{jn} \)) is substituted into the second-stage model. Then, the relaxed second-stage model is solved to obtain the objective value \( H_{jn}^{R} \). If \( U^{j} + H_{jn}^{R} \geq Z_{UB} \), then the feasible solution is discarded. Otherwise, the second-stage model is solved to obtain the corresponding objective value of \( H_{jn}^{R} \). If necessary, the upper and lower bounds of the first-stage objective value are updated before proceeding to Step 2.

In this algorithm, the lower bound of the objective value of the second-stage model is used, which is the value of the corresponding relaxed model. This is achieved by removing all the integer constraints. In other words, the second-stage model (a MIP model) is transformed into a linear programming (LP) model, with the advantage that the LP model can be solved efficiently. The objective value of the relaxed model should be the lower bound of the objective value of the second-stage model, by which it is possible to avoid some unnecessary computations.

**B. REMOVING SYMMETRIC SOLUTIONS**

This algorithm actually tests some feasible solutions of the first-stage model. However, it is worth noting that all of the ORs are the same, which results in numerous symmetric solutions. When the number of ORs increases, the number of symmetric solutions increases exponentially. Then, it is possible to obtain all the solutions of the first-stage model, including the symmetric solutions. However, if the corresponding second-stage model of the symmetric solutions is solved, then it results in repetitive computations, which are unnecessary and makes the model unsolvable. Hence, the symmetric solutions should be removed. Moreover, this paper applies a rule to standardize the OR allocations and defines the term set level, i.e., the highest level of a surgeon within a set of surgeons. The rule is that a set of surgeons with a higher set level should be allocated to ORs that are denoted with smaller numbers. This rule is illustrated in the following example.

**Example 1:** For example, there are 3 surgeries and 2 ORs. Two possible allocation scenarios are denoted as \( H_{1} \) and \( H_{2} \), and

\[
H_{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_{2} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}
\]

According to this rule, \( H_{1} \) is legal, whereas \( H_{2} \) is illegal. In \( H_{1} \), the set level in OR 1 is 1, which is higher than the set level in OR 2, i.e., 2. For \( H_{2} \), the set level in OR 1 is 2, which is lower than the set level in OR 1, i.e., 1, and thus, it is illegal.

Next, some constraints are proposed to remove the symmetric solutions to ensure that all the solutions conform to the aforementioned rule. After defining binary variable \( v_{ir} \) \((i \in I \text{ and } r \in R)\), let

\[
v_{ir} = h_{ir} \quad \forall r \in R \tag{59}
\]

\[
v_{ir} \geq v_{i-1,r} \quad 2 \leq i \leq I, \quad \forall r \in R \tag{60}
\]

\[
v_{ir} \geq h_{ir} \quad \forall i \in I, \quad r \in R \tag{61}
\]

\[
v_{ir} \leq v_{i-1,r} + h_{ir} \quad 2 \leq i \leq I, \quad \forall r \in R \tag{62}
\]

\[
v_{i,r-1} \geq v_{ir} \quad \forall i \in I, \quad 2 \leq r \leq R \tag{63}
\]

Constraint (59) initializes \( v_{ir} \), while Constraints (60) and (61) make \( v_{i-1,r} = 1 \), if \( h_{ir} = 1 \) or \( v_{i-1,r} = 1 \). Constraint (62) ensures that \( v_{ir} = 0 \), if \( h_{ir} = 0 \) and \( v_{i-1,r} = 0 \), while Constraint (63) clarifies that the allocation conforms to the rule.

In sum, this section discussed how to solve the two-stage model. A bound-based algorithm was proposed to solve the model. In order to avoid any unnecessary computations, a rule was proposed to remove the symmetric solutions. We investigate the performance of our proposed algorithm in Section V.

**V. NUMERICAL STUDIES**

This section introduces the data collected and investigates the performance of the proposed algorithm. The sensitivities of some of the parameters are also analyzed.

**A. DATA COLLECTION AND PARAMETER SETTINGS**

For this paper, the data was collected from the largest department of a hospital in China, i.e., the department of thoracic surgery. There was a total of 2,706 observations (surgeries) performed by six surgeons in the department. The surgery durations greatly depended on the surgery type. The surgeries were categorized into 11 types. The mean and standard...
deviation of the surgery duration of each surgery type are obtained.

This paper tested the performance of the proposed algorithm using the surgery duration data. We set $T = 600$ (i.e., 10 hours). The OR opening cost per day was set at $c^R = 2000$, while the overtime cost was set at $c^O = 20$.

The cost of working as an assistant and the cost of waiting time per minute depended on the level of the surgeons. It is reasonable to assume that the cost for $L_2$ surgeons is generally higher than that for $L_3$ surgeons. Thus, this paper set $c^A_m = 4 + 0.1(M - m + 1)$, if $m \in L_2$, and $c^A_m = 1 + 0.1(A - m + 1)$, if $m \in L_3$. For example, if $L_2 = 3$, then the cost of working as an assistant to a $L_2$ surgeon is $4.3, 4.2, 4.1$; and if $L_3 = 3$, the cost of working as an assistant to a $L_3$ surgeon is $1.3, 1.2, 1.1$. Similarly, this paper set $c^W_m = 20$, if $m \in L_1$, $c^W_m = 15$, if $m \in L_2$, $c^W_m = 10$, and if $m \in L_3$. The experiments are solved by using the Gurobi Python interface on a Window 7, Intel i5, 4 GB RAM computer.

**B. INITIALIZATION OF THE ALGORITHM**

This section utilizes several examples to illustrate how the proposed algorithm works and how to initialize the solution of the first-stage model in Step 1. In this case, the set included one $L_1$ surgeon, three $L_2$ surgeons, and one $L_3$ surgeon. Each of the $L_1$ and $L_2$ surgeons performed two surgeries per day. First, a simple initialization of the first-stage of the algorithm, i.e., the lower bound of the Step 1 model, $Y^{bd} = 0$, was implemented. The first 100 iterations are summarized in Table 4.

<table>
<thead>
<tr>
<th>Start Setting</th>
<th>Update No.</th>
<th>No. Solns.</th>
<th>Cost 1</th>
<th>Cost 2</th>
<th>Total Cost</th>
<th>Upper Bound</th>
<th>Time elapsed(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR = 1</td>
<td>1</td>
<td>1</td>
<td>3432.20</td>
<td>12990</td>
<td>16422.20</td>
<td>16422.20</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 1</td>
<td>20</td>
<td>1</td>
<td>3815.00</td>
<td>21670</td>
<td>25485.00</td>
<td>14025.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 1</td>
<td>40</td>
<td>1</td>
<td>3872.40</td>
<td>15715</td>
<td>19587.40</td>
<td>14025.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 1</td>
<td>60</td>
<td>1</td>
<td>3919.50</td>
<td>14940</td>
<td>18859.50</td>
<td>14025.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 1</td>
<td>80</td>
<td>1</td>
<td>3946.30</td>
<td>20845</td>
<td>24791.30</td>
<td>14025.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 1</td>
<td>100</td>
<td>1</td>
<td>3968.30</td>
<td>20375</td>
<td>24343.30</td>
<td>14025.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 2</td>
<td>20</td>
<td>7</td>
<td>5432.20</td>
<td>10200</td>
<td>15632.20</td>
<td>14570.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 2</td>
<td>40</td>
<td>7</td>
<td>5872.40</td>
<td>8620</td>
<td>14435.00</td>
<td>11040.19</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 2</td>
<td>60</td>
<td>7</td>
<td>5919.50</td>
<td>4260</td>
<td>10179.50</td>
<td>10024.80</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 2</td>
<td>80</td>
<td>7</td>
<td>5946.30</td>
<td>7485</td>
<td>13431.30</td>
<td>10024.80</td>
<td>115.3</td>
</tr>
<tr>
<td>OR = 2</td>
<td>100</td>
<td>7</td>
<td>5968.30</td>
<td>6150</td>
<td>12118.30</td>
<td>10024.80</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 1</td>
<td>1</td>
<td>14</td>
<td>7009.00</td>
<td>5793</td>
<td>12804.00</td>
<td>12804.00</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>4</td>
<td>7</td>
<td>7010.20</td>
<td>1890</td>
<td>8900.20</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>20</td>
<td>14</td>
<td>7015.90</td>
<td>-</td>
<td>'&gt;UB'</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>40</td>
<td>7</td>
<td>7023.50</td>
<td>7490</td>
<td>14513.50</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>60</td>
<td>7</td>
<td>7029.90</td>
<td>5425</td>
<td>12454.00</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>80</td>
<td>21</td>
<td>7034.30</td>
<td>1890</td>
<td>8924.30</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>100</td>
<td>14</td>
<td>7039.00</td>
<td>3075</td>
<td>10114.00</td>
<td>8900.20</td>
<td>115.3</td>
</tr>
<tr>
<td>LB = 7000, OR = 2</td>
<td>285</td>
<td>14</td>
<td>7070.00</td>
<td>785</td>
<td>7855.90</td>
<td>7855.90</td>
<td>2411.1</td>
</tr>
</tbody>
</table>
by setting the number of opened ORs and giving a rational bound to the first-stage model.

C. SENSITIVITY ANALYSIS

1) COST EFFECTS

This section discusses how the costs in the second stage influence the allocation decisions in the first-stage. First, it is important to investigate how the cost of working as an assistant in L3 surgeries affects surgeon assignments. In this regard, the example in the last subsection will be used. By increasing the cost of working as an assistant to L3 surgeons, the model is solved in order to obtain the assignment decisions (see Table 5). In general, the costs seem to influence surgery allocations and surgeon assignments. However, according to Table 5, an L3 surgeon is more likely to be assigned as an assistant when the costs increase and when the OR allocation changes, which is related to the cost in the first stage.

Next, the cost effects of L3 surgeon’s waiting time costs are examined (see Table 6). When the waiting time costs increase, surgery allocations and surgeon assignments also change. Moreover, when the costs increase, a surgeon is more likely to be assigned to surgeries that start early.

2) EFFECTS OF ROBUSTNESS

This section focuses on how the robustness setting influences performance. By changing the value of \( \delta \) (see Section III-E for its definition), the allocation and start time results are obtained, which makes it possible to generate random durations and run simulations. The simulation results are illustrated in Table 7. The first column is within the range of \( \delta \), under which random data is generated. From the third to the seventh columns, the simulation results are presented, including the average costs in the two stages. The numbers, 0, 0.1, \( \cdots \), are the values of \( \delta \) used when solving the two-stage model. Moreover, the first part of Table 7 shows the results when the durations are uniformly distributed from the smallest to the largest values. The result is the best when \( \delta = 0.3 \). In the second part of Table 7, when the durations are close to the average, the best performance occurs when \( \delta = 0 \), e.g., the allocation decisions are made by the average value of the durations. Furthermore, when the durations are much larger or smaller than the average value, \( \delta = 0.3 \) is again the best. Finally, the expected the first-stage cost decreases when the larger value of \( \delta \) is used. Based on these findings, performance is generally better if robustness is considered. However, how much the robustness should be considered depends on each case.

D. POLICY COMPARISON

This subsection compares three policies. The first policy is to use the results of the proposed two-stage model. The second policy is an easy-to-implement policy, based on the results in the former examples (i.e., Tables 5 and 6). In this policy, L1 surgeons are allocated to different ORs as much as possible to ensure that the senior surgeons can start early. In addition, the L2 surgeons, they are allowed to work in the same OR as assistants for one another. For example, there are two surgeons in the same OR, and the second surgeon is the assistant of the first surgeon, and vice versa. Through this type of arrangement, waiting time can be avoided or at least significantly reduced. In order to illustrate the benefits of this easy-to-implement policy, a myopic policy is implemented in which the allocation decisions are made by simply minimising the costs in the first stage. The three aforementioned policies are presented as follows:

**Policy I:** (Optimal policy) Allocate surgeries and assign surgeons to work as assistants, according to the results of the proposed two-stage model.

**Policy II:** (Easy-to-implement policy)

*Step 1:* Allocate L1 surgeons to different ORs as much as possible.

*Step 2:* Allocate L2 surgeons to the ORs and avoid overtime as much as possible.

*Step 3:* Assign L3 surgeons as assistants to L1 surgeons as much as possible. If necessary, assign L2 surgeons to L1 surgeons.

*Step 4:* Assign surgeons in the same OR to work as assistants for one another.

**Policy III:** (Myopic policy) Allocate surgeries and assign surgeons to work as assistants, according to the results of minimising the first-stage costs. The start time is determined by solving the corresponding second-stage model.

---

**TABLE 5. Cost effects of L3 surgeons working as assistants.**

<table>
<thead>
<tr>
<th>Surgery</th>
<th>( C^A = 1.1 )</th>
<th>( C^A = 2.1 )</th>
<th>( C^A = 3.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR a</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
</tr>
<tr>
<td>OR a</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
</tr>
</tbody>
</table>

**TABLE 6. Cost effects of L3 surgeon waiting time.**

<table>
<thead>
<tr>
<th>Surgery</th>
<th>( C^W = 5 )</th>
<th>( C^W = 10 )</th>
<th>( C^W = 15 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR a</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
</tr>
<tr>
<td>OR a</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
<td>1 3 1 5 2 5 1</td>
</tr>
</tbody>
</table>

**TABLE 7. Effects of robustness.**

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>cost</td>
<td>8201.16</td>
<td>8155.96</td>
<td>8055.46</td>
<td>8003.84</td>
<td>7938.56</td>
</tr>
<tr>
<td>cost 2</td>
<td>1334.36</td>
<td>1213.18</td>
<td>1172.67</td>
<td>1280.28</td>
<td>1704.98</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>9535.52</td>
<td>9368.24</td>
<td>9626.13</td>
<td>9284.12</td>
<td>9643.54</td>
<td></td>
</tr>
<tr>
<td>0-0.5</td>
<td>cost</td>
<td>7841.87</td>
<td>7819.93</td>
<td>7644.48</td>
<td>7670.34</td>
<td>7602.33</td>
</tr>
<tr>
<td>cost 2</td>
<td>1548.83</td>
<td>1562.98</td>
<td>930.29</td>
<td>1524.17</td>
<td>2249.54</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>8366.52</td>
<td>8862.93</td>
<td>8623.77</td>
<td>9194.72</td>
<td>9581.86</td>
<td></td>
</tr>
<tr>
<td>0.5-1</td>
<td>cost</td>
<td>5582.16</td>
<td>5350.77</td>
<td>8400.69</td>
<td>8371.90</td>
<td>8294.26</td>
</tr>
<tr>
<td>cost 2</td>
<td>2045.21</td>
<td>1900.17</td>
<td>1790.80</td>
<td>1110.01</td>
<td>1449.11</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>10607.37</td>
<td>10333.74</td>
<td>10161.29</td>
<td>9481.91</td>
<td>9743.37</td>
<td></td>
</tr>
</tbody>
</table>
A large example is utilised to highlight the performance of the policies, in which there are five ORs, three L1 surgeons, seven L2 surgeons, and three L3 surgeons, with each surgeon performing two surgeries per day, i.e., $L_1 = 3$, $L_2 = 7$, and $L_3 = 3$. That is, there are totally 20 surgeries performed in a day, which is large enough for the daily needs in the department. In this case, robustness is neglected, the durations are randomly generated, and 1,000 simulations are run to obtain the average cost in both stages (see Table 8). The costs obtained by implementing the three policies as well as the corresponding gap are listed in Table 8. The gap is calculated by the following formula: $\text{gap} = \frac{\text{cost(Policy III)}}{\text{cost(Policy I)}} - 1$. The optimal policy shows the best performance, i.e., the corresponding average cost is the smallest. Overall, to some extent, it is still acceptable, although the cost is higher than that of Policy I by 13.47%. In Policy II, although waiting time is avoided as much as possible, there is no optimisation due to the combination of surgeries in the ORs. Moreover, it is possible to determine the benefits of using the easy-to-implement policy by comparing it with the myopic policy. In sum, the easy-to-implement policy is acceptable, especially when attempting to avoid waiting time.

VI. CONCLUSIONS AND FUTURE WORK

This paper focuses on resource allocation and surgery scheduling problems with consideration of assistant surgeons. A two-stage model is proposed, a bound-based algorithm is developed to solve the model, and a method is presented to avoid symmetry solutions in the first-stage model. The findings show that the model and algorithm are useful for resolving the aforementioned issues in ORs. Results of the sensitivity analysis indicate that the costs, including waiting time costs and the costs of working as an assistant, significantly influence the decisions in the first stage. Based on the allocation results of several examples, the easy-to-implement policy is acceptable, especially when attempting to avoid/reduce waiting time. Moreover, the gap between the cost of using the easy-to-implement policy and that of using the optimal policy is also acceptable.

Finally, one possible direction for future research would be to consider surgeon preferences and the performance of such collaborations. More specifically, a surgeon might prefer working with a particular assistant, which can result in better performance, e.g., shorter surgery durations or higher surgery quality. Hence, such an assistant should be assigned to each surgeon, which obviously impacts the surgery scheduling problem. Additionally, due to the complexity of the models, we handle the uncertainties of surgery durations in a simple way to make the model tractable. That is, the stochastic model is transformed to being a deterministic model by using the simple $l_1$-norm uncertain set. It would be an interesting direction to handle the uncertainties in a more sophisticated way.

REFERENCES

J. Wang et al.: Two-Stage Approach for Resource Allocation and Surgery Scheduling With Assistant Surgeons


JIN WANG received the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, in 2015. He is currently an Assistant Professor with the College of Management, Shenzhen University. His research interests include optimal modeling and scheduling approaches with applications in complexed healthcare and/or manufacturing systems, such as rehabilitation treatments and operation rooms, automated manufacturing systems, cluster tools, and robotic cells.

JING CHU received the master’s degree in medicine from Jinan University, Guangzhou, China. He was a Visiting Scholar with Vivantes Auguste Viktoria Hospital, Germany. He is currently the Deputy Chief Physician in urology and the Deputy Director of the Medical Department, Zhuhai People’s Hospital. He is also the Head of the Urology Group and the Deputy Director of the Robotic Surgery Group, Zhuhai People’s Hospital. He focuses on minimally invasive diagnosis and prostate hyperplasia, prostate cancer, and diagnosis and andrological diseases.

XIN LI received the B.S. degree in industrial engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, the M.S. degree in industrial engineering and logistics management from Shanghai Jiao Tong University, Shanghai, China, in 2011, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2014.

KWOK-LEUNG TSUI was a Professor with the School of Industrial and Systems Engineering, Georgia Institute of Technology. He is currently the Chair Professor of industrial engineering with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, and the Founder and the Director of the Centre for Systems Informatics Engineering. His current research interests include data mining, surveillance in healthcare and public health, prognostics and systems health management, calibration and validation of computer models, process control and monitoring, and robust design and Taguchi methods. He is also a Fellow of the American Statistical Association, the American Society for Quality, and the International Society of Engineering Asset Management, and a U.S. representative to the ISO Technical Committee on Statistical Methods. He was a recipient of the National Science Foundation Young Investigator Award. He was the Chair of the INFORMS Section on Quality, Statistics, and Reliability and the Founding Chair of the INFORMS Section on Data Mining.

* * *