Surgery Scheduling in the Presence of Operating Room Eligibility and Dedicated Surgeon: An Adaptive Composite Dispatching Method

Shan Wang[a], Huiqiao Su[b], Guohua Wan[c] and Liwei Zhong[d]

[a]School of Business, Sun Yat-sen University, Guangzhou, China; [b]Huawei Technologies Co Ltd, Shanghai, China; [c]Antai College of Economics and Managements, Shanghai Jiao Tong University, Shanghai, China; [d]Shanghai Municipal Hospital of Traditional Chinese Medicine Affiliated to Shanghai University of Traditional Chinese Medicine, Shanghai, China

ARTICLE HISTORY

Compiled July 19, 2024

ABSTRACT

Motivated by a real problem in a big hospital in China, we study a daily surgery scheduling problem with operating room eligibility and dedicated surgeon. We model the problem as a parallel machine scheduling problem with machine and resource constraints to minimize the makespan, and innovatively propose an adaptive composite dispatching method to deal with such a strongly NP-hard problem. The dispatching rule is a combination of three popular rules LPT, LFJ and LRW, each of which can deal with some special features of the scheduling problems, and the scaling parameters are estimated through a statistical model learned from historical data. The adaptive composite dispatching method is easy-to-implement, fast, adaptable, robust and flexible. To examine the performance of the proposed solution approach, we first carry out a series of computational experiments showing that the adaptive composite dispatching method works very well compared to the optimal solution. Using a real data set, we further conduct a case study showing that our solution approach can improve the current practice by significantly shortening the makespan and reducing the overnight work.

KEYWORDS

Healthcare; Surgery Scheduling; Composite Dispatching Rule; Data-driven Modeling; Machine Eligibility; Resource Constraint

CONTACT Guohua Wan Email: ghwan@sjtu.edu.cn, or Liwei Zhong Email: zhongliwei1972@163.com

1. Introduction

With the increasing demand for healthcare services resulted from the aging population and rising living standards, healthcare systems are facing great challenges worldwide (WHO 2017). As a major source of revenue as well as the cost center of a medical institution, surgical services play critical roles in healthcare services in hospitals. As reported in Muñoz, Muñoz III, and Wise (2010), about 30% of the total health expenditure is surgical service-related cost. Therefore, efficient utilization of the costly and limited surgical resources is critical in controlling healthcare costs and providing good patient care. In past decades, researchers and practitioners have devoted much efforts in developing surgery scheduling systems to improve surgical services, however, its performance in practices is still under expectation. In fact, surgery scheduling is very complicated and challenging due to various constraints and special requirements (Abdelrasol, Harraz, and Eltawil 2014).

Eligibility of operating rooms is one of the most important constraints in surgery scheduling. Some operating rooms are configured with high-tech equipment which may cost millions of dollars, while some operating rooms are only of basic functions (Vairaktarakis and Cai 2003). Depending on the operating room configurations, surgeries may only be performed in specific types of operating rooms. Such a constraint largely increases the difficulty in surgery scheduling. Without careful management, advanced operating rooms may be assigned too many surgeries (as some surgeries can only be performed in these operating rooms and some unnecessary surgeries are assigned to them as well), thus excessive overtimes occur.

Dedicated surgeons are another important constraint in surgery scheduling. In most scenarios, a surgery can only be performed by a dedicated surgeon (Wang, Su, and Wan 2015). If such a constraint is ignored, the scheduled time slots of the two surgeries requiring the same surgeon may be overlapped, thus one of them has to be delayed until the surgeon is available again. Consequently, the utilization of operating rooms decreases and the overtime increases.

Despite the significance and prevalence of operating room eligibility and dedicated surgeons, the current practice of surgery scheduling undervalues or even ignores these critical constraints. An interview with the deputy director of a comprehensive hospital in Shanghai, China, reveals that when an initial surgery schedule is generated (commonly by first come first service rule), the scheduler manually adjusts it to satisfy various constraints. Among those constraints, operating room eligibility and dedicated surgeon are main hurdles in finalizing the schedule. In the north one of the hospital's two campuses, there are 22 operating rooms for in-patient elective surgeries, around 150 surgeons, and almost 500 types of surgeries. In 2017, around 20,000 elective surgeries are performed in total (more than 40 surgeries per working day). To increase the utilization of operating rooms and reduce the overtime, it is urged to balance workload among all operating rooms and shorten the overtime working of surgeons, leading to the objective to minimize the makespan of the surgery schedule. The South campus of the hospital has different configurations of operating rooms and different characteristics of patients population, thus developing a scheduling approach which can be easily adapted to a new environment is also an important concern.

Motivated by both practical and theoretical challenges, we leverage the historical data and develop a heuristic approach for solving such a challenging surgery scheduling problem, taking into considerations of operating room eligibility and dedicated surgeons. We model the surgery scheduling problem as a parallel machine scheduling problem with machine eligibility and resource constraints. Specifically, surgeries are regarded as jobs, operating rooms are modeled as parallel machines on which jobs are processed, and surgeons are modeled as resources (throughout the paper, we may use "surgery" and "job", "operating room" and "machine", "surgeon" and "resource" interchangeably). We then propose adaptive composite dispatching method to deal with such problem. This approach is innovative as the scaling parameters are estimated through a statistical model learned from historical data. The performance and robustness of the adaptive composite dispatching method are demonstrated by extensive computational experiments. Furthermore, a case study is conducted with a real data set to show how our solution approach performs in practice.

Our contributions can be summarized as follows. We innovatively propose the first adaptive composite dispatching method in surgery scheduling in which the scaling parameters are estimated through a statistical model learned from historical data. The advantages of such solution approach include its effectiveness and efficiency in dealing with large-scale surgery scheduling problem, its adaptability to changing environments, its robustness in different settings and its flexibility in handling complex constraints. Moreover, the adaptive composite dispatching method is user-friendly, and managerial insights can be extracted through the solution processes. Although the solution approach is motivated by surgery scheduling in healthcare, it can be extended to production and service operations scheduling where machines/severs have eligibility constraint and jobs need dedicated resource.

The remainder of this paper is organized as follows. In Section 2, we review the related literature. In Section 3, we illustrate the problem and the model. Section 4 describes how to develop the composite dispatching rule by learning the statistical model which estimates scaling parameters from historical data. We present extensive computational experiments and case study with real data in Section 5 and Section 6, respectively. Section 7 concludes our work and discusses the future research in this vein.

2. Literature Review

The literature on surgery scheduling ranges from strategical level to operational level (May et al. 2011). This study is closely related to the studies at the operational level (see, e.g., Cardoen, Demeulemeester, and Beliën 2010, Guerriero and Guido 2011, Abdelrasol, Harraz, and Eltawil 2014, Samudra et al. 2016, Zhu et al. 2019, for detailed reviews). There are two types of surgery – elective surgery and emergency surgery. For elective surgery, practitioners can schedule them well in advance since the surgical information is available; while for emergency case, scheduling must be done in online fashion, i.e., the surgery is scheduled upon patient's arrival. In practice, management normally reserves specific operating rooms or time slots for emergency surgery (please refer to Van Riet and Demeulemeester 2015, Kahraman and Topcu 2018, Duma and Aringhieri 2019, Jung et al. 2019, for more details). In the hospital with which we collaborate, several operating rooms are dedicated to emergency surgery, thus we focus on scheduling elective surgery in this study.

In the literature on elective surgery scheduling at operational level, some papers (see, e.g., Fei et al. 2008, Day, Garfinkel, and Thompson 2012, Roshanaei et al. 2017, Zhang et al. 2020, Guo et al. 2021) focus on assignment of specific surgeons and time blocks of operating rooms to surgeries within a planning horizon; while some (see, e.g., Pham and Klinkert 2008, Cardoen, Demeulemeester, and Beliën 2009a, Herring and Herrmann 2012) study how to determine the sequence of the surgeries. Our study is different from theirs in the sense that we consider both assignment and sequencing decisions. There are also some studies concerning both assignment and sequencing decisions. For example, Jebali, Alouane, and Ladet (2006) propose a two-step approach to deal with the surgery scheduling problems, where the first step is to assign the surgeries to the operating rooms and the second step is to sequence them. Batun et al. (2011) present a two-stage stochastic mixed-integer programming model for the integrated surgery scheduling and sequencing problem, where the surgeries are assigned to the operating rooms first, and then are sequenced. Wang, Su, and Wan (2015) consider a surgery scheduling problem with various constraints to minimize the makespan of the schedule, and develop decomposition-based heuristic algorithms to solve the assignment and sequencing problem separately. Our study is different from theirs as we simultaneously solve the assignment and sequencing problems and can generate better solutions. Hashemi Doulabi, Rousseau, and Pesant (2016) also consider simultaneous decisions of assignment and sequencing by developing a branch-and-price-and-cut algorithm, but operating room eligibility is not considered in their model.

For the constraints in surgery scheduling, researchers are mostly concerned with personnel constraints (e.g. Roland et al. 2010, Di Martinelly, Baptiste, and Maknoon 2014, Rath, Rajaram, and Mahajan 2017, Bandi and Gupta 2020), preference constraints (see Samudra et al. (2016) for more discussions), and coordination constraints (e.g., Min and Yih 2010, Day, Garfinkel, and Thompson 2012, Xiao et al. 2018). However, operating room constraints receive little attention (Zhao and Li 2014). To the best of our knowledge, only Wang, Su, and Wan (2015) propose a heuristic which can handle both personnel constraints (e.g., surgeon dedication) and operating room constraints (e.g., operating room eligibility). In our study, we take both of them into account.

In terms of solution approaches, mathematical programming models and algorithms (e.g., Roland et al. 2010, Day, Garfinkel, and Thompson 2012, Vijayakumar et al. 2013, Hashemi Doulabi, Rousseau, and Pesant 2016), simulations (e.g., Lamiri et al. 2008, Min and Yih 2010, Zhang and Xie 2015) and heuristic algorithms (e.g., Lamiri, Grimaud, and Xie 2009, Creemers, Beliën, and Lambrecht 2012, Choi and Wilhelm 2014) are widely used. In our study, we model the surgery scheduling problem as a machine scheduling problem with complicated constraints (see, e.g., Pham and Klinkert 2008, Wang, Su, and Wan 2015) and innovatively develop the first adaptive composite dispatching method in which the scaling parameters are estimated through a statistical model learned from historical data. And we will see, this approach is adaptable, robust, flexible, and effective.

Dispatching rules are simple, easy to implement and of low computational complexity. It is very useful especially for large-scale complicated scheduling problems (Morton and Pentico 1993). For example, Bahaji and Kuhl (2008) evaluate dispatching rules and other scheduling policies in wafer fabrication facilities, and point out that the composite dispatching rules are robust and effective. Recently, composite dispatching rules are widely used for dealing with complicated scheduling problems. For example, Su, Pinedo, and Wan (2017) propose a composite dispatching rule for parallel machine scheduling with nested eligibility constraints; Ying, Lin, and Lu (2017) develop a dynamic dispatching rule for single-machine scheduling problem with a common due window; Li, Freiheit, and Miao (2017) design two new dispatching rules for flow shop scheduling problem; Lee (2018) studies dispatching rules for a real-life production scheduling problem originating from a plant producing Acrylonitrile-Butadiene-Styrene (ABS) plate products. Our study is the first one using statistical learning method to estimate the scaling parameters in the ranking index of a composite dispatching rule. Since historical data is employed to derive the dispatching rule, our approach is adaptive to different instances and able to be easily updated with changing environment. It is worth noting that Jun, Lee, and Chun (2019) also use statistical learning models to design dispatching rules. They learn the dispatching rule directly, while we choose to learn the scaling parameters.

3. Problem Description

Consider a scheduling problem with I operating rooms in parallel and J surgeries that have to be assigned to these operating rooms and then scheduled. Surgery j can only be processed in an operating room that belongs to a specific subset of operating rooms, denoted by $\mathbb{M}_j$, and operating rooms in set $\mathbb{M}_j$ are indifferent for processing surgery j; the number of operating rooms in set $\mathbb{M}_j$ is M_j, i.e., $M_j = |\mathbb{M}_j|$. Following Vairaktarakis and Cai (2003), $\mathbb{M}_j$ can be described by the *Eligibility Matrix* $\mathcal{A}$, where the element $\mathcal{A}[i,j]$ represents whether operating room i is able to process surgery j

$$\mathcal{A}[i,j] = \begin{cases} 1 & \text{operating room } i \text{ is eligible to process surgery } j, \\ 0 & \text{otherwise.} \end{cases}$$

Assume further there are K surgeons and surgery j can only be processed by a predetermined surgeon, denoted by R_j. (Table A1 in Appendix summarizes all notations appearing in this paper.)

Following Cardoen, Demeulemeester, and Beliën (2009b), Day, Garfinkel, and Thompson (2012), Vijayakumar et al. (2013), Di Martinelly, Baptiste, and Maknoon (2014), Wang, Su, and Wan (2015), Hashemi Doulabi, Rousseau, and Pesant (2016), we assume that the processing time of surgery j is deterministic, denoted by p_j. Although the processing time may be stochastic, schedules are often generated with the estimated processing times in practice. Using the estimated processing time as p_j can simplify the whole scheduling problem, and, by applying our solution approach in stochastic environment, we can still achieve good schedules with acceptable optimality gap (see Section 5.4 for details).

The objective of our model is to find a feasible schedule that minimizes the makespan, denoted by C_{max}, i.e., the time when all surgeries are completed. Smaller makespan can be translated to higher utilization of operating rooms, less overtime working of surgeons, and earlier completion times for patients, leading to lower overall cost.

Adopting notations in Pinedo (2016), this model can be described as $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$ (if we treat operating rooms as machines, surgeries as jobs and surgeons as resources). Here Pm denotes a scheduling environment of parallel machines (operating rooms), $\mathbb{M}_j$ indicates machine eligibility constraint and any machine in $\mathbb{M}_j$ is indifferent to job j, while any machine not in $\mathbb{M}_j$ cannot process job (surgery) j. Res_{K11} denotes the resource (surgeon) constraint – there are K resources in total, each resource has 1 unit, and each job requires only 1 unit of the resource. C_{max} is the makespan of the schedule, and the objective is to minimize this makespan.

The problem $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$ is a generalization of the problem $Pm \mid\mid C_{max}$, a well-known strongly NP-Hard problem (Pinedo 2016), thus it is NP-Hard in strong sense as well. For a special case where resource constraint is ignored, i.e., $Pm \mid \mathbb{M}_j \mid C_{max}$, Shchepin and Vakhania (2005) develop a polynomial-time algorithm with a worst-case ratio $2 - 1/m$, and Vairaktarakis and Cai (2003) develop a branch-and-bound algorithm which can solve it to optimality with up to 50 jobs. For our problem with resource constraint, i.e., $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$, it is very difficult to develop a polynomial-time algorithm with performance guarantee. In fact, its complexity can be examined by its MILP (Mixed Integer Linear Programming) formulation below.

$$\min_{x_{ij},\ y_{jj'},\ s_j, C_{max}} \quad C_{max} \tag{1}$$

$$\sum_{i=1}^{I} x_{ij} = 1 \quad \text{for all } j \tag{2}$$

$$y_{jj'} + y_{j'j} = 1 \quad \text{for all } j,\ j' \tag{3}$$

$$x_{ij} \leq \mathcal{A}[i, j] \quad \text{for all } i,\ j \tag{4}$$

$$B(2 + y_{jj'} - x_{ij} - x_{ij'}) + s_j \geq p_{j'} + s_{j'} \quad \text{for all } i,\ j,\ j' \tag{5}$$

$$By_{jj'} + s_j \geq p_{j'} + s_{j'} \quad \text{for all } j,\ j' \text{ with } R_j = R_{j'} \tag{6}$$

$$C_{max} \geq s_j + p_j \quad \text{for all } j \tag{7}$$

$$x_{ij},\ y_{jj'} \in \{0, 1\} \quad \text{for all } i,\ j,\ j' \tag{8}$$

In this model, x_{ij} is a binary variable representing whether to schedule surgery j in operating room i or not, $y_{jj'}$ is a binary variable denoting whether surgery j is before surgery j' or not, s_j denotes starting time of surgery j, and B is a big number. Constraint (2) means that a surgery can only be processed in one operating room. Constraint (3) requires that surgery j is either before or after surgery j'. Constraint (4) is the eligibility constraint. Constraints (5) and (6) indicates that if surgery j is after surgery j' and they are scheduled in the same operating room or they are processed by same surgeon, then the starting time of surgery j cannot be earlier than the completion time of surgery j'. Now, instead of developing an exact algorithm to solve this complicated model, we turn to a heuristic approach and develop a composite dispatching rule.

4. Adaptive Composite Dispatching Method

A Dispatching rule is a general procedure that can be used in dealing with practical scheduling problems and implemented relatively easily in industrial scheduling systems (Pinedo 2016). A composite dispatching rule is a ranking expression that combines a number of elementary rules, where the key is to choose the scaling parameters to properly adjust the contribution of each rule to the total ranking index (Pinedo 2016).

In our adaptive composite dispatching method, historical data is used to train the statistical model which estimates the scaling parameters. For a new instance (an instance consists of surgeries to be scheduled during a day and the corresponding information about the operating rooms and surgeons), according to the generated ranking index with the predicted scaling parameters, the surgeries are dispatched. To design a composite dispatching rule which can deal with our problem $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$, we consider three elementary rules as follows.

(1) Longest Processing Time first (LPT) rule,
(2) Least Flexible Job first (LFJ) rule
(3) Largest Resource Workload first (LRW) rule

LPT is a classic heuristic for scheduling problems which minimize makespan; LFJ is designed for machine eligibility; and LRW is a resource-based rule for resource-constraint scheduling problem (Pinedo 2016, Błażewicz et al. 2007, Morton and Pentico 1993). Thus, our dispatching rule, which is a proper combination of these three rules, is able to take the objective and all constraints into consideration. We call it Adaptive Triple-L (ATL) rule.

In the following, we first introduce the definition of *ranking index*, then we discuss how to determine the scaling parameters.

4.1. *Ranking Index*

When an operating room becomes available, the surgery with the highest ranking index will be dispatched to it for operations (in case of ties, arbitrarily pick one surgery). Let $\mathbb{U}(i,t)$ denote the set of surgeries which can be processed in operating room i and have not been dispatched at time t. The ranking index of a surgery $j \in \mathbb{U}(i,t)$ for operating room i at time t, $I_j(i,t)$, is computed as

$$I_j(i,t) = \frac{p_j}{\mathcal{P}} \exp\left(-\frac{M_j}{\sigma_1 \overline{M}(i,t)}\right) \exp\left(\frac{\mathcal{P}(R_j,t)}{\sigma_2 \overline{\mathcal{P}}(i,t)}\right), \tag{9}$$

where p_j is the processing time of surgery j; $\mathcal{P}$ is the total processing time of all surgeries in the instance, i.e.,

$$\mathcal{P} = \sum_{j=1}^{n} p_j; \tag{10}$$

$M_j = |\mathbb{M}_j|$ is the number of operating rooms which are eligible for surgery j; $\overline{M}(i,t)$ is the average value of M_j for surgeries that can be processed in operating room i and have not been dispatched at time t, i.e.,

$$\overline{M}(i,t) = \frac{\sum_{j \in \mathbb{U}(i,t)} M_j}{\sum_{j \in \mathbb{U}(i,t)} 1}; \tag{11}$$

$\mathcal{P}(R_j, t)$ is the total processing time of un-dispatched surgeries of surgeon R_j at time t, i.e.,

$$\mathcal{P}(R_j, t) = \sum_{j' \in \cap_i \mathbb{U}(i,t) \text{ and } R_{j'} = R_j} p_{j'}, \tag{12}$$

and $\overline{\mathcal{P}}(i,t)$ is the average value of $\mathcal{P}(R_j, t)$ for all surgeries which can be processed in operating room i but have not been dispatched at time t, i.e.,

$$\overline{\mathcal{P}}(i,t) = \frac{\sum_{j \in \mathbb{U}(i,t)} \mathcal{P}(R_j, t)}{\sum 1_{\{j \in \mathbb{U}(i,t)\}}}. \tag{13}$$

Note that σ_1 and σ_2 are scaling parameters to be determined, which significantly affect the performance of the composite dispatching rule (Lin, Fowler, and Pfund 2013). As we mentioned before, *ATL* rule is a hybrid of 3 classic elementary rules – LPT, LFJ and LRW. Depending on the value of σ_1 and σ_2, the corresponding *ATL* rule becomes several special cases shown in Table 1.

Table 1. Special Cases of *ATL* with Different Scaling Parameters

	Small σ_1	Medium σ_1	Large σ_1
Small σ_2	LFJ+LRW	LRW	LRW
Medium σ_2	LFJ	LPT+LFJ+LRW	LPT+LRW
Large σ_2	LFJ	LPT+LFJ	LPT

[a] *LPT*: longest processing time first;
[b] *LFJ*: least flexible job first;
[c] *LRW*: largest remaining workload of resource first.

The smaller σ_1 is, the larger the impact of LFJ rule is. Similarly, when σ_2 decreases, *LRW* rule becomes important. When both of them increase, *LPT* rule dominates. Thus these two scaling parameters are critical and *ATL* rule is governed by them.

Moreover, different instances have different characteristics thus may need different scaling parameters. For example, if the eligibility constraint is tight, then *LFJ* rule should play an important role in the composite dispatching rule; and if the workload of surgeon is very light, we should not consider *LRW* rule. So for each instance, we need to determine specific scaling parameters which are suitable for it. To determine the value of these two parameters, we turn to an adaptive approach.

4.2. *Determining Scaling Parameters σ_1 and σ_2*

Before discussing how to determine σ_1 and σ_2, we first introduce a series of factors for characterizing a problem instance.

The factor φ, called *surgery flexibility*, is used to describe the overall flexibility of all surgeries in a given instance. It is defined as

$$\varphi = 1 - \sum_{j=1}^{J} \frac{p_j}{M_j} / \mathcal{P}, \tag{14}$$

where p_j is the processing time of surgery j, $\mathcal{P}$ is the total processing time of all surgeries, and M_j is the number of operating rooms that are eligible for surgery j. Larger φ implies more flexible surgeries, i.e., they can be processed in more operating rooms. If each surgery can be processed in any operating room, i.e., $\mathbb{M}_i = \{1, 2, ..., I\}$, then $\varphi = 1 - 1/I$.

The second factor ψ, called *operating room flexibility*, is defined as

$$\psi = (\sum_{i=1}^{I} \sum_{j \in \mathbb{J}_i} p_j)/(I * \mathcal{P}), \tag{15}$$

where the set $\mathbb{J}_i$ is the set of surgeries that can be processed in operating room i. Similar to φ, ψ stands for the overall flexibility of operating rooms. Large ψ indicates that operating rooms are flexible, i.e., they are able to process many types of surgeries. If every operating room can process all surgeries, i.e., $\mathbb{J}_i = \{1, 2, ..., J\}$, then $\psi = 1$.

The third factor, which plays an important role in makespan minimization of a scheduling problem (Vairaktarakis and Cai 2003), is to measure the flexibility of the eligibility matrix $\mathcal{A}$. It is determined by the distribution of "1"s in $\mathcal{A}$ (whose element is either 1 or 0). We capture the number of "1"s in $\mathcal{A}$ using the *process flexibility index*, denoted by ρ and is defined as

$$\rho = \frac{\sum_{i,j} \mathcal{A}[i,j] - J}{J(I-1)}. \tag{16}$$

Without loss of generality, we assume that, for each surgery, there exists at least one operating room eligible to process it. Without this assumption, there is not a feasible schedule. Thus we have $0 \leq \rho \leq 1$.

The fourth factor ϕ, called *surgeon workload balance*, is defined as

$$\phi = \frac{\min\{\sum_{R_j=1} p_j, \sum_{R_j=2} p_j, ..., \sum_{R_j=K} p_j\}}{\max\{\sum_{R_j=1} p_j, \sum_{R_j=2} p_j, ..., \sum_{R_j=K} p_j\}}. \tag{17}$$

This factor is to describe the workload balance of surgeons in an instance. If all surgeons have similar workload, then ϕ is close to 1.

The next step is to build a model which links scaling parameters with these four factors. Given the historical data, i.e., a set of instances which are used to train the model, we can first employ the so-called "Sequential Uniform Design" method (Fang and Lin 2003) to search for the best pair of σ_1 and σ_2 for each instance (the details can be found in Appendix B). And then a proper statistical learning model can be built to estimate the relationship $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$. For example, we can use the following linear models,

$$\sigma_1 = \alpha_0 + \alpha_1\varphi + \alpha_2\psi + \alpha_3\rho + \alpha_4\phi, \tag{18}$$

$$\sigma_2 = \beta_0 + \beta_1\varphi + \beta_2\psi + \beta_3\rho + \beta_4\phi. \tag{19}$$

In such a case, model specification or variable selection methods, for instance, $Step-wise\ Forward\ Regression$ or $Step-wise\ Backward\ Regression$, may be used to find the best regression model. Other statistical learning models, e.g., regression tree, random forest and neural network, are also proper to learn the relationship $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$.

To summarize, the framework of our adaptive composite dispatching method can be described in Algorithm 1.

4.3. *Discussions on the Adaptive Composite Dispatching Method*

The advantages of the proposed adaptive composite dispatching method are as follows. First, the scaling parameters in the ranking index are adaptive to different instances,

Algorithm 1 Adaptive Composite Dispatching Method

```
1: for each instance in the training data set do
2:     Search for the best pair of $(\sigma_1, \sigma_2)$ by Algorithm 3 (in Appendix)
3:     Calculate the factors $(\varphi, \psi, \rho, \phi)$ by (14),(15), (16) and (17)
4: end for
5: Choose a proper statistical learning model to estimate the relationship $(\sigma_1, \sigma_2) \sim$
   $(\varphi, \psi, \rho, \phi)$
6: Given a new instance, calculate $(\varphi, \psi, \rho, \phi)$ and predict the corresponding $(\hat{\sigma}_1, \hat{\sigma}_2)$
   by the estimated model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$.
   Apply $ATL$ rule according to ranking index (9).
```

so it outperforms traditional composite dispatching rules with fixed scaling parameters. Second, the solution approach is flexible. It can be modified to incorporate more elementary rules to deal with other constraints and objectives. For example, if we want to take the uncertainty of the processing time into account, we can use the variance of processing time to construct one factor to characterize an instance. Third, historical data is utilized to learn the statistical model which estimates the scaling parameters, so this method is adaptive to the environment changes with updated data. Last but not least, managerial insights can be obtained from scaling parameters. For example, if the estimated σ_1 and σ_2 are very large, then for this instance, the processing time is the main hurdle to shorten the makespan, while the constraints which relate to operating rooms and surgeons are less important.

The limitation of our solution approach is that *ATL* rule can only generate active schedules, i.e., operating rooms cannot be idle if there is any available unscheduled surgery; while the optimal schedule may let operating rooms strategically idle. The reason that we only consider active schedules is that it is more humanitarian since patients may feel panic and anxious if they cannot be served when all resources are available for services. Moreover, the computational experiments in the following section show that the performance of the active schedule is very close to the optimal schedule.

5. Computational Experiments with Small-scale Synthetic Instances

For small-scale instances, we are able to find the optimal schedule by solving the MILP problem (see the Model (1)-(8) in Section 3) via off-the-shell solvers such as CPLEX or Gurobi, thus we can compare the schedule generated by ATL rule with the optimal schedule to examine the performance of our solution approach. To do this, we first generate 160 instances of 4 operation rooms, 8 surgeons and 16 surgeries as training set. The eligibility matrix $\mathcal{A}$, the resource requirement vector $\{R_1, ..., R_j, ..., R_J\}$ and the processing time vector $\{p_1, ..., p_j, ..., p_J\}$ are randomly drawn from identical distribution. For each instance in training set, we first search for the best scaling parameters σ_1 and σ_2 (see Algorithm 3 in Appendix B). In the sequential uniform design, by taking both of effectiveness and efficiency into account and comparing 16 different combinations of hyper-parameters, we finally set $B = 10$, $n^{(t)} = 30$ for $\forall t$, $q = 30$, $\epsilon = 0.001$ and $\beta = 0.5$. It is worth note that, comparing to the optimal schedule which is obtained by solving MILP Model (1)-(8) via Gurobi 9.0.2, the final schedule generated by ATL rule with scaling parameters obtained by sequential uniform design performs 1.3% worse on average. This indicates that sequential uniform design is capable to obtain a near optimal schedule. With the searched scaling parameters and the calculated four factors of 160 instances in the training set, we can build the model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$ using statistical learning method. Here we use decision tree with proper pruning by 3-fold cross validation as the statistical learning model[1].

5.1. *Performance on Training Set*

For each instance in the training set, we get the estimated scaling parameters $(\hat{\sigma}_1, \hat{\sigma}_2)$ with the fitted model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$. With $(\hat{\sigma}_1, \hat{\sigma}_2)$, we compute the ranking index in ATL rule, and obtain the corresponding schedule. Comparing with the optimal schedule solved by MILP model, the average performance gap is 1.91% and the maximum performance gap is 5.77%. The optimality gap is defined as

$$\frac{V(\mathit{Optimal\ Schedule}) - V(\mathit{ATL\ Schedule})}{V(\mathit{Optimal\ Schedule})}.$$

where $V(x)$ represents the makespan of a schedule. The overall performance gap is resulted from two aspects. First, as discussed before, ATL rule only generates active schedule, so there exists gap between the best active schedule and the optimal schedule (such gap is about 1.3% on average for these 160 instances). Second, the difference between the estimated $(\hat{\sigma}_1, \hat{\sigma}_2)$ and the best pair of (σ_1, σ_2) also result in performance

[1]We also consider linear regression model, and it turns out that decision tree achieves larger R^2.

gap. To better illustrate these two types of gaps, we take one instance as an example (see Table 2).

Table 2. Details of Instance No.26

Surgery ID	Processing Time	Dedicated Surgeon ID	Eligible Operating Room ID
1	20	7	1, 2, 3
2	9	7	1, 2, 3, 4
3	20	6	2, 3
4	19	2	2,3,4
5	19	4	1,2,3
6	20	3	3,4
7	7	5	2,3
8	11	5	4
9	13	2	1,3,4
10	11	7	3,4
11	16	2	1,3,4
12	5	3	3,4
13	20	4	1
14	18	6	1,2,4
15	19	6	1,4
16	20	1	1,2,3,4

Three schedules generated for Instance No. 26 are shown in Figure 1. The first one is the optimal schedule solved by MILP model. We can see that, when $t = 36$, i.e., when surgery 11 is completed, there are surgery 4, 7, 12 and 13 available for operating room 3. However, instead of dispatching these 4 surgeries immediately, the optimal schedule chooses to wait one time slot, and dispatches surgery 3 when surgeon 6 completes surgery 14. The second schedule is the schedule generated by ATL rule with the best scaling parameters. Since ATL rule can only generate active schedule, the corresponding makespan is 2 slots longer than the optimal schedule. Such gap is the first type of gaps mentioned before. The second type of gaps is the gap between the second schedule and the third schedule. The second schedule is generated with the best pair of scaling parameters, while the third schedule is generated with the estimated ones. Since they are different, the third schedule is worse than the second one.

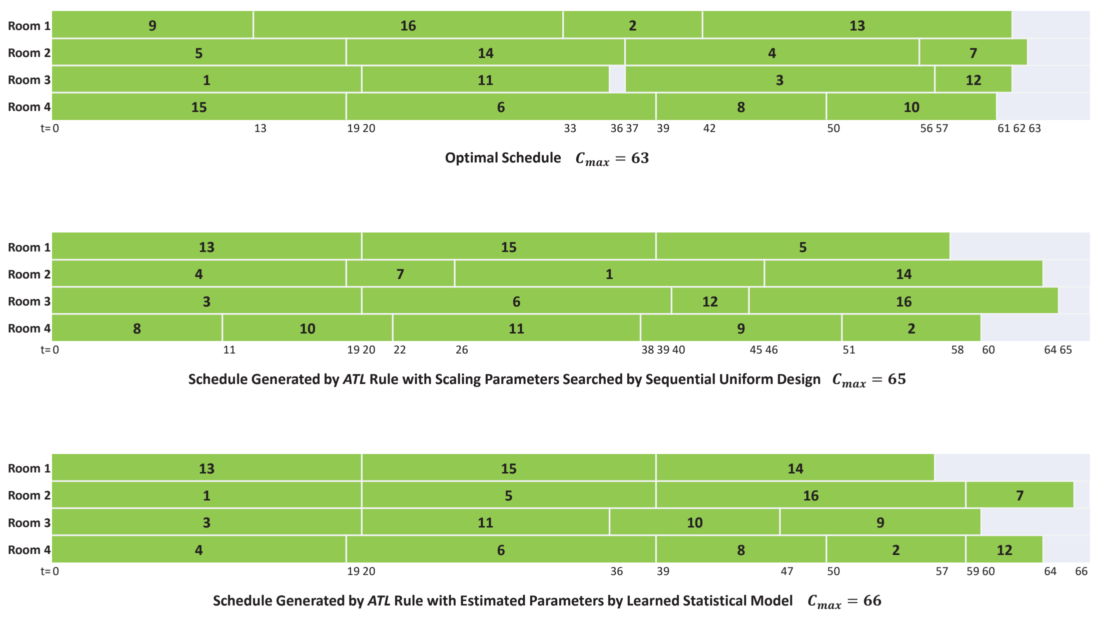

Figure 1. Schedule Comparison for Instance No.26

Alt Text: The first figure shows the optimal schedule. After the completion of surgery 11, room 3 is strategically idle for 1 time slot. The second figure shows the schedule generated by ATL rule with the best scaling parameters. And the third schedule shows the schedule generated by ATL rule with estimated scaling parameters. There is no strategical idling in schedules generated by ATL rule.

5.2. *Performance on Test Set from Same Distribution*

To further examine the performance of the proposed adaptive composite dispatching method, we generate 40 instances as test set I. The eligibility matrix $\mathcal{A}$, the resource requirement vector $\{R_1, ..., R_j, ..., R_J\}$ and the processing time vector $\{p_1, ..., p_j, ..., p_J\}$ are randomly drawn from same distribution as the training set. For each instance in test set I, we first calculate the factors $(\varphi, \psi, \rho, \phi)$, and estimate the scaling parameters with the learned model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$. After applying $(\hat{\sigma}_1, \hat{\sigma}_2)$ into the ranking index of *ATL* rule, we obtain a schedule and the corresponding makespan. Comparing to the optimal schedule obtained by solving the MILP model, we found that the average optimality gap is 3.17% and the maximum optimality gap is 9.43%. As a dispatching rule, *ATL* rule works very well when the relationship $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$ is learned from right population.

5.3. *Performance on Test Set from Different Distribution*

It is possible that the new instance is quite different from the instances which are used in training. To examine the robustness of ATL rule, we generate another 40 instances as test set II. The eligibility matrix $\mathcal{A}$, the resource requirement vector $\{R_1, ..., R_j, ..., R_J\}$ and the processing time vector $\{p_1, ..., p_j, ..., p_J\}$ are randomly drawn from *different* distribution from the training set. Following the same experiment procedure as for test set I, we generate a schedule and the corresponding makespan for each instance in test set II. Compared to the optimal schedule, the schedule generated by ATL rule achieves 4.42% average performance gap and 16.4% maximum performance gap. Considering that test set II and training set are from different populations, we believe that the average performance is good and the worst performance is acceptable.

5.4. *Performance in Stochastic Environment*

In our model, we assume that the processing time of a surgery is deterministic. However, as mentioned before, the prediction of processing time may not be accurate and may have deviation from the realized one. To check the performance of the adaptive composite dispatching method in stochastic environment, we generate test set III of 40 instances which is the same as test set I, except for that the actual processing time of surgery j is p_j plus or minus a random error term. Then we can see how the proposed method performs if we only know the mean of the processing time. For each instance, we first estimate the scaling parameters with the mean processing time p_j, and generate the schedule by ATL rule (note that such schedule is exactly as same as the one generated in Section 5.2). Applying the generated schedule on the instance with actual (realized) processing time, we obtain the actual makespan. Compared to the the optimal schedule obtained by solving the MILP model with realized processing time, we compute the optimality gap, which represents the regret of applying our method in stochastic environment. Finally, we find that the average gap (regret) is 4.81% and the maximum gap (regret) is 15.63%, indicating that our method can still achieve good performance even if the processing time is not deterministic.

Table 3 summarizes the results of our computational experiments for small-scale synthetic instances.

Table 3. Performance Gap between the schedule generated by *ATL* Rule and Optimal Schedule

	Training Set	Test Set I	Test Set II	Test Set III
Average Gap(%)	1.91	3.17	4.42	4.81
Maximum Gap(%)	5.77	9.43	16.39	15.63

[a]Training Set is of 160 instances from same distribution;

[b]Test Set I is of 40 instances from same distribution;

[c]Test Set II is of 40 instances from different distribution;

[d]Test Set III is of 40 instances with stochastic processing times.

6. Case Study with Real Data Set

We conduct a case study based on a real data set to evaluate the performance of our adaptive composite dispatching method for practical implementation.

6.1. *Data Description*

From the collaborating hospital, a large comprehensive one in Shanghai, China, we obtain 180-day data of elective surgeries from January 2, 2017 to June 30, 2017 in its north campus. We regard them as 180 instances (the data in one day is an instance). The data set contains totally 22 operating rooms for elective surgeries. After consulting with the practitioners and cleaning the raw data properly, we obtain the eligibility constraint of operating rooms, (i.e., the specific sets of operating rooms that can process certain types of surgeries), and the details of each instance, including the starting times, the surgery processing times and the responsible surgeons. In this hospital, the current surgery scheduling policy is FCFS (First Come First Served). From the data we notice that the total number of surgeries in a day varies a lot, from the smallest number, 1 surgery only, to the largest number, 103 surgeries. We also observe that there are usually more than 40 surgeries on weekdays and less than 10 surgeries on the weekends or holidays. To obtain the appropriate estimating model for σ_1 and σ_2, we use the 117 instances with more than 40 surgeries. The reasons for such selection are as follows. First, it is relatively easy to make a schedule when there are less than 40 surgeries in a day. Second, it is more beneficial to make a good schedule when there are more surgeries to be scheduled. Third, from the practical point of view, the problem faced by the hospital is to schedule surgeries when the operating rooms and surgeons face heavy workload. In the end, we also use the excluded 63 instances to do robustness check.

We split the data set of 117 selected instances into a training set with 77 instances and a test set of the remaining 40 instances. In each instance, there are 22 operation rooms, i.e., $I = 22$; the number of surgeries is more than 40, i.e., $J > 40$; the total workload and the number of surgeons are different (observed directly from the data).

6.2. *Training the Statistical Model for Scaling Parameters*

For each instance in training set, we first search for the best scaling parameters σ_1 and σ_2. The whole procedure follows sequential uniform design in Appendix B (for details please refer to Algorithm 3). As for the hyper-parameters in the sequential uniform design, here we set $n^1 = 37$, $n^t = 31$ for $t \geq 2$, $q = 30$, $B = 10$, $\beta = 0.5$ and $\epsilon = 0.001$. Finally we obtain the best pair of σ_1 and σ_2 for each instance.

From these 77 pairs of σ_1 and σ_2, we find that σ_1 is changing with the instances, while σ_2 is quite stable (5.296) for most instances. For efficiency and convenience, we set $\sigma_2 = 5.296$ as a fixed number. And for σ_1, we adopt linear regression to train the statistical model $\sigma_2 \sim (\varphi, \psi, \rho, \phi)$, and use $Step-wise\ Backward\ Regression$ to do model specification. The results suggest that there is no significant fixed effect and only φ and ρ should be included in the model. Thus we have the following models for $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$:

$$\sigma_1 = 12.68\varphi - 12.20\rho \tag{20}$$

$$\sigma_2 = 5.296 \tag{21}$$

6.3. *Improvement by* **ATL** *rule*

With the statistical model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$, we can estimate the scaling parameters $(\hat{\sigma}_1, \hat{\sigma}_1)$, calculate the ranking index, and generate schedule by ATL rule for each instance in test set. Next, we examine the performance of ATL rule.

To measure the performance improvement of ATL over current practices (the real schedules) in the hospital, we introduce a ratio RI (*Relative Improvement*), which is defined as

$$\text{RI} = \frac{V(\textit{Real Schedule}) - V(\textit{ATL Schedule})}{V(\textit{Real Schedule})},$$

where $V(x)$ represents the makespan of a schedule. Similarly, we define RD (*Relative Deviation*) to measure the performance gap between ATL schedule and the real schedule.

$$\text{RD} = \frac{V(\textit{Real Schedule}) - V(\textit{ATL Schedule})}{V(\textit{ATL Schedule})}.$$

We perform four evaluations of the average of RI (ARI) and the average of RD (ARD) to obtain the whole picture of performance comparisons. From Table 4, we see that the ARI is more than 15% and the ARD is up to 21% on test set (Evaluation 1); unsurprisingly, ATL rule performs much better on the training set plus test set where the ARI is up to 20% and the ARD is almost 35% (Evaluation 2).

As discussed before, we do not include the instances with less than 40 surgeries into the training set for learning the model $(\sigma_1, \sigma_2) \sim (\varphi, \psi, \rho, \phi)$. However, when we test the ATL rule on these small instances, the ARI is 75.18% and the ARD is as high as 592.37% (Evaluation 3). Moreover, the ARI is 38.95% and the ARD is 229.59% when we test all the 180 instances (Evaluation 4). We then carefully examine the data, and find that the real schedules for these 63 instances are very loose, indicating that practitioners were not concerned with the idleness of the operating rooms once a feasible schedule without overtime is obtained. As shown in our experiments, the situation can be improved a lot with our adaptive composite dispatching method.

Table 4. Average RI and Average RD for ATL Schedule and Real Schedule

	Evaluation 1	Evaluation 2	Evaluation 3	Evaluation 4
ARI(%)	15.27	19.58	75.18	38.95
ARD(%)	21.41	34.37	592.37	229.59

[a]Evaluation 1 represents the evaluation on the testing set (40 instances);
[b]Evaluation 2 represents the evaluation on the training set + testing set (117 instances);
[c]Evaluation 3 represents the evaluation on the instances which have less than 40 surgeries in a day (63 instances);
[d]Evaluation 4 represents the evaluation on all 180 instances.

This hospital also suffers excessive overtime working of surgeons. According to the hospital regulations, the regular working time is between 8:00am and 5:00pm on weekdays. We then check the instances that surgeons can go off work on time. The data shows that in only 10 out of 180 days, or 5.56% of the instances, surgeons can be off duty on time; the worst thing is that, in 55 days, or 30.56% of instances, surgeons complete all surgeries in the early morning next day. Apparently, it is an undesirable situation for surgeons as well as the hospital management. After applying our solution, we find that among all 180 instances, the number of instances in which all surgeons complete their work within the regular working time is increased from 10 to 80 (from 5.56% to 39%); and only in 19 instances, surgeons have to work overnight, decreased from 30.56% to 10.56%. By applying the proposed method, the serious situation faced by surgeons and the hospital management can be improved significantly.

Comparing *ATL* schedule with the real schedule shows the impressive performance of our adaptive composite dispatching method. However, it is not clear how good *ATL* schedule is. To address this concern, we conduct the following numerical study.

6.4. *Comparing with Near-optimal Schedules*

For 117 instances with more than 40 surgeries, it is almost impossible to obtain the optimal schedule within reasonable time. Here we use SA (Simulated Annealing) algorithm to obtain the near-optimal schedule by limiting the total computational time (Johnson et al. 1989). Matsuo, Juck Suh, and Sullivan (1989) point out that a good seed solution may considerably decrease the computational time when solving hard scheduling problems. Therefore, we use *ATL* schedule as the seed solution. Such treatment has two benefits. First, we are able to obtain a schedule which is closer to the optimal one in a reasonable time. Second, we can check that how much improvement can be achieved by additional effort (through SA algorithm).

Following the scheme of SA algorithm in Lee and Pinedo (1997), we propose a Modified Simulated Annealing (*MSA*) algorithm, designed specifically for our surgery scheduling problem $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$.

Step 1 is the initialization step. There are 100 stages and each stage has 1000 iterations. In each iteration, Step 6 randomly generates a new schedule by moving the first surgery to the second position, the second surgery to the third position, and the third surgery to the first position, and check if the new schedule is feasible and better

Algorithm 2 Modified Simulated Annealing (MSA) Algorithm

```
 1: $ST = 100$: maximum stages; $IT = 1000$: maximum iterations at each stage;
    $p_a = 0.0495$: acceptance probability;
    $Stage = 1$: current stage number; $Iteration = 1$: current iteration number;
    Let $s$ be the schedule generated bye $ATL$ rule, and $s$ denotes the current schedule;
    $C_{max} = V(s)$: current makespan of schedule $s$
 2: while $Stage \leq ST$ do
 3:   $Stage = Stage + 1$; $p_a = 0.05 - 0.05 * Stage/100$; $Iteration = 1$
 4:   while $Iteration \leq IT$ do
 5:     $Iteration = Iteration + 1$
 6:     Compute a new schedule $s'$ in the following way:
        Randomly pick 3 surgeries in the schedule and sequentially change the position
        of these 3 surgeries
 7:     if $s'$ is feasible then
 8:       if $V(s') < C_{max}$ then
 9:         $s = s'$; $C_{max} = V(s')$
10:       else
11:         Generate a random number $\tilde{r}$ from the uniform distribution in $[0, 1]$
12:         if $\tilde{r} < p_a$ then
13:           $Iteration = IT + 1$; $s = s'$; $C_{max} = V(s')$
14:         end if
15:       end if
16:     end if
17:   end while
18: end while
```

than the current one. If so, update the current schedule (Step 7-8). In Step 10-12, in order to escape from the local optimal solution, there is a probability of p_a to jump to the next stage, allowing one to consider an inferior solution as a candidate. The acceptance probability p_a, which is generally positive, decreases when the number of stages increases, meaning that the probability of escaping from a local optimal solution at a later stage is smaller (it becomes 0 at the last stage). This linearly decreasing form of p_a follows Matsuo, Juck Suh, and Sullivan (1989) and Vakharia and Chang (1990). (There are also distinct forms of acceptance probability p_a, see Metropolis et al. (1953), Matsuo, Juck Suh, and Sullivan (1989) and Vakharia and Chang (1990) for more discussions.)

Applying Algorithm 2 on 117 instances of the real data set with more than 40 surgeries, we calculate the improvement achieved by Modified SA Algorithm. Numerical results are summarized in Table 5. The first row is the relative improvement which is calculated by

$$\frac{V(\mathit{ATL\ Schedule}) - V(\mathit{MSA\ Schedule})}{V(\mathit{MSA\ Schedule})},$$

where $V(x)$ represents the makespan of a schedule. The second row is the absolute improvement, i.e.,

$$V(\mathit{ATL\ Schedule}) - V(\mathit{MSA\ Schedule}).$$

From the table, we can see that the average improvement is only 7 minutes and the maximum improvement is less than half an hour. We can conclude that it is difficult to obtain significant improvement by applying Modified SA Algorithm, and the schedule is of high quality and very close to the optimal schedule.

Table 5. Improvement Achieved by Modified SA Algorithm

	Minimum	Average	Median	Maximum
Relative Improvement (%)	0	1.27	1.06	4.82
Absolute Improvement (minutes)	0	7	6	25

[a]The number of instances is 117;

[b]There are 31 instances that cannot be improved;

[c]There are only 11 instances having over 15-minute improvement.

For evaluating the practical implications, we also collect some feedback from the collaborating hospital and find that our proposed method not only provides a tool for automating the surgical scheduling process but also improves the schedules by 10-20% on average in the past two years.

7. Conclusions and Discussions

In this study, we innovatively propose an adaptive composite dispatching method to solve the surgery scheduling problem with operating room eligibility and dedicated surgeon. We first model the problem as a parallel machine scheduling problem with machine eligibility and resource constraint, i.e., $Pm \mid \mathbb{M}_j, Res_{K11} \mid C_{max}$. By estimating the scaling parameters in ranking index through statistical model learned from the historical data, we design an adaptive composite dispatching rule – ATL rule. The computational experiments and real-data case study show that our method is able to achieve good and robust performance.

To the best of our knowledge, this is the first adaptive composite dispatching method utilized in surgery scheduling, which is proven to be effective, efficient, robust and flexible. And this approach is innovative in the sense that the scaling parameters, which are used to combine different elementary rules, are estimated through a statistical model learned from historical data. As a dispatching rule, ATL rule is adaptive to different characteristics of the instances thus it is able to generate proper schedule case by case. When the environment changes, it is easy to use new data to update the statistical model. Moreover, it can be easily modified to combine more elementary dispatching rules to deal with other constraints and objectives. And for practitioners, by examining the estimated scaling parameters, they are able to find the bottleneck of current surgical systems. Although this work is motivated by surgery scheduling in healthcare, the solution approach can be extended to any service and production contexts with server/machine eligibility and dedicated resource constraint.

There are some issues for implementation. First, choosing a proper statistical learning model is critical to the performance of ATL rule. A model which is too simple may be under-fitted, while a complicated one may be over-fitted. Cross validation can help a lot in model specification. Second, when the environment is changing, updating the learned statistical model in time is important. Such updates can be conducted periodically or even in on-line fashion – when new data is available, sample some instances from it, search for the best scaling parameters, update the training set with the sampled instances and the corresponding best scaling parameters, and retrain the statistical learning model. In this way, the proposed method can be easily applied to changing environment.

For future research direction, it will be interesting to look into different constraints, objectives, and environments. First, hospitals are paying more attentions to the surgery scheduling problems with pre-operative and post-operative services, thus upstream and downstream constraints need to be considered in the model. Second, the proposed adaptive composite dispatching method is designed under the assumption of deterministic processing time. Though it works well even in stochastic environment, it is worth to consider processing time variability in developing the ranking index and estimating the scaling parameters. Third, it will be interesting to consider how to schedule emergency surgeries by applying composite dispatching rule in online fashion. Last, operating room eligibility is an important and relevant constraint in surgical scheduling practice. It is valuable to design efficient algorithms with theoretical performance guarantees for such challenging problems.

Data Availability Statement: The data used in Section 6 contains the patient's privacy information thus cannot be shared publicly. However, part of the data is available upon request.

References

Abdelrasol, Z., N. Harraz, and A. Eltawil. 2014. "Operating Room Scheduling Problems: a Survey and a Proposed Solution Framework." *Transactions on Engineering Technologies* 717–731.

Bahaji, N., and M. E. Kuhl. 2008. "A Simulation Study of New Multi-Objective Composite Dispatching Rules, CONWIP, and Push Lot Release in Semiconductor Fabrication." *International Journal of Production Research* 46 (14): 3801–3824.

Bandi, C., and D. Gupta. 2020. "Operating Room Staffing and Scheduling." *Manufacturing & Service Operations Management* 22 (5): 958–974.

Batun, S., B. T. Denton, T. R. Huschka, and A. J. Schaefer. 2011. "Operating Room Pooling and Parallel Surgery Processing under Uncertainty." *INFORMS Journal on Computing* 23 (2): 220–237.

Błażewicz, J., K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. 2007. *Handbook on Scheduling: from Theory to Applications*. Springer Science & Business Media.

Cardoen, B., E. Demeulemeester, and J. Beliën. 2009a. "Optimizing a Multiple Objective Surgical Case Sequencing Problem." *International Journal of Production Economics* 119 (2): 354–366.

Cardoen, B., E. Demeulemeester, and J. Beliën. 2009b. "Sequencing Surgical Cases in a Day-Care Environment: an Exact Branch-and-Price Approach." *Computers & Operations Research* 36 (9): 2660–2669.

Cardoen, B., E. Demeulemeester, and J. Beliën. 2010. "Operating Room Planning and Scheduling: A Literature Review." *European Journal of Operational Research* 201 (3): 921–932.

Choi, S., and W. E Wilhelm. 2014. "On Capacity Allocation for Operating Rooms." *Computers & Operations Research* 44: 174–184.

Creemers, S., J. Beliën, and M. Lambrecht. 2012. "The Optimal Allocation of Server Time Slots over Different Classes of Patients." *European Journal of Operational Research* 219 (3): 508–521.

Day, R., R. Garfinkel, and S. Thompson. 2012. "Integrated Block Sharing: A Win–Win Strategy for Hospitals and Surgeons." *Manufacturing & Service Operations Management* 14 (4): 567–583.

Di Martinelly, C., P. Baptiste, and M. Y. Maknoon. 2014. "An Assessment of The Integration of Nurse Timetable Changes with Operating Room Planning and Scheduling." *International Journal of Production Research* 52 (24): 7239–7250.

Duma, D., and R. Aringhieri. 2019. "The Management of Non-Elective Patients: Shared vs. Dedicated Policies." *Omega* 83: 199–212.

Fang, K. T., and D. K. J. Lin. 2003. "Ch. 4. Uniform Experimental Designs and Their Applications in Industry." *Handbook of Statistics* 22: 131–170.

Fei, H., C. Chu, N. Meskens, and A. Artiba. 2008. "Solving Surgical Cases Assignment Problem by a Branch-and-Price Approach." *International Journal of Production Economics* 112 (1): 96–108.

Guerriero, F., and R. Guido. 2011. "Operational Research in the Management of the Operating Theatre: a Survey." *Health Care Management Science* 14 (1): 89–114.

Guo, C., M. Bodur, D. M. Aleman, and D. R. Urbach. 2021. "Logic-Based Benders Decomposition and Binary Decision Diagram Based Approaches for Stochastic Distributed Operating Room Scheduling." *INFORMS Journal on Computing* 33 (4): 1551–1569.

Hashemi Doulabi, S. H., L. M. Rousseau, and G. Pesant. 2016. "A Constraint-Programming-Based Branch-and-Price-and-Cut Approach for Operating Room Planning and Scheduling." *INFORMS Journal on Computing* 28 (3): 432–448.

Herring, W. L., and J. W. Herrmann. 2012. "The Single-Day Surgery Scheduling Problem: Sequential Decision-Making and Threshold-Based Heuristics." *OR Spectrum* 34 (2): 429–459.

Jebali, A., A. B. H. Alouane, and P. Ladet. 2006. "Operating Rooms Scheduling." *International Journal of Production Economics* 99 (1-2): 52–62.

Johnson, D. S., C. R. Aragon, L. A. McGeoch, and C. Schevon. 1989. "Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning." *Operations Research* 37 (6): 865–892.

Jun, S., S. Lee, and H. Chun. 2019. "Learning Dispatching Rules Using Random Forest in Flexible Job Shop Scheduling Problems." *International Journal of Production Research* 57 (10): 3290–3310.

Jung, K. S., M. Pinedo, C. Sriskandarajah, and V. Tiwari. 2019. "Scheduling Elective Surgeries with Emergency Patients at Shared Operating Rooms." *Production and Operations Management* 28 (6): 1407–1430.

Kahraman, C., and Y. I. Topcu. 2018. *Operations Research Applications in Health Care Management.* Springer.

Lamiri, M., F. Grimaud, and X. Xie. 2009. "Optimization Methods for a Stochastic Surgery Planning Problem." *International Journal of Production Economics* 120 (2): 400–410.

Lamiri, M., X. Xie, A. Dolgui, and F. Grimaud. 2008. "A Stochastic Model for Operating Room Planning with Elective and Emergency Demand for Surgery." *European Journal of Operational Research* 185 (3): 1026–1037.

Lee, C. H. 2018. "A Dispatching Rule and a Random Iterated Greedy Metaheuristic for Identical Parallel Machine Scheduling to Minimize Total Tardiness." *International Journal of Production Research* 56 (6): 2292–2308.

Lee, Y. H., and M. Pinedo. 1997. "Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times." *European Journal of Operational Research* 100 (3): 464–474.

Li, W., T. Freiheit, and E. Miao. 2017. “A Lever Concept Integrated with Simple Rules for Flow Fhop Fcheduling.” *International Journal of Production Research* 55 (11): 3110–3125.

Lin, Y. K., J. W. Fowler, and M. E. Pfund. 2013. “Multiple-Objective Heuristics for Scheduling Unrelated Parallel Machines.” *European Journal of Operational Research* 227 (2): 239–253.

Matsuo, H., C. Juck Suh, and R. S. Sullivan. 1989. “A Controlled Search Simulated Annealing Method for the Single Machine Weighted Tardiness Problem.” *Annals of Operations Research* 21 (1): 85–108.

May, J. H., W. E. Spangler, D. P. Strum, and L. G. Vargas. 2011. “The Surgical Scheduling Problem: Current Research and Future Opportunities.” *Production and Operations Management* 20 (3): 392–405.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. “Equation of State Calculations by Fast Computing Machines.” *The Journal of Chemical Physics* 21 (6): 1087–1092.

Min, D., and Y. Yih. 2010. “Scheduling Elective Surgery under Uncertainty and Downstream Capacity Constraints.” *European Journal of Operational Research* 206 (3): 642–652.

Morton, T. E., and D. W. Pentico. 1993. *Heuristic Scheduling Systems: with Applications to Production Systems and Project Management.* Vol. 3. John Wiley & Sons.

Muñoz, E., W. Muñoz III, and L. Wise. 2010. “National and Surgical Health Care Expenditures, 2005–2025.” *Annals of Surgery* 251 (2): 195–200.

Pham, D. N., and A. Klinkert. 2008. “Surgical Case Scheduling as a Generalized Job Shop Scheduling Problem.” *European Journal of Operational Research* 185 (3): 1011–1025.

Pinedo, M. L. 2016. *Scheduling: Theory, Algorithms, and Systems (Fifth Edition).* Springer.

Rath, S., K. Rajaram, and A. Mahajan. 2017. “Integrated Anesthesiologist and Room Scheduling for Surgeries: Methodology and Application.” *Operations Research* 65 (6): 1460–1478.

Roland, B., C. Di Martinelly, F. Riane, and Y. Pochet. 2010. “Scheduling an Operating Theatre under Human Resource Constraints.” *Computers & Industrial Engineering* 58 (2): 212–220.

Roshanaei, V., C. Luong, D. M. Aleman, and D. R. Urbach. 2017. “Collaborative Operating Room Planning and Scheduling.” *INFORMS Journal on Computing* 29 (3): 558–580.

Samudra, M., C. Van Riet, E. Demeulemeester, B. Cardoen, N. Vansteenkiste, and F. E. Rademakers. 2016. “Scheduling Operating Rooms: Achievements, Challenges and Pitfalls.” *Journal of Scheduling* 19 (5): 493–525.

Shchepin, E. V., and N. Vakhania. 2005. “An Optimal Rounding Gives a Better Approximation for Scheduling Unrelated Machines.” *Operations Research Letters* 33 (2): 127–133.

Su, H., M. Pinedo, and G. Wan. 2017. “Parallel Machine Scheduling with Eligibility Constraints: A Composite Dispatching Rule to Minimize Total Weighted Tardiness." *Naval Research Logistics* 64 (3): 249–267.

Vairaktarakis, G. L., and X. Cai. 2003. "The Value of Processing Flexibility in Multipurpose Machines." *IIE Transactions* 35 (8): 763–774.

Vakharia, A. J., and Y. L. Chang. 1990. "A Simulated Annealing Approach to Scheduling a Manufacturing Cell." *Naval Research Logistics* 37 (4): 559–577.

Van Riet, C., and E. Demeulemeester. 2015. "Trade-Offs in Operating Room Planning for Electives and Emergencies: A Review." *Operations Research for Health Care* 7: 52–69.

Vijayakumar, B., P. J. Parikh, R. Scott, A. Barnes, and J. Gallimore. 2013. "A Dual Bin-Packing Approach to Scheduling Surgical Cases at a Publicly-Funded Hospital." *European Journal of Operational Research* 224 (3): 583–591.

Wang, S., H. Su, and G. Wan. 2015. "Resource-Constrained Machine Scheduling with Machine Eligibility Restriction and its Applications to Surgical Operations Scheduling." *Journal of Combinatorial Optimization* 30 (4): 982–995.

WHO. 2017. *10 Facts on Ageing and the Life Course.* https://www.who.int/news-room/fact-sheets/detail/10-facts-on-ageing-and-health, Accessed February 18, 2022.

Xiao, G., W. van Jaarsveld, M. Dong, and J. van de Klundert. 2018. "Models, Algorithms and Performance Analysis for Adaptive Operating Room Scheduling." *International Journal of Production Research* 56 (4): 1389–1413.

Ying, K. C., S. W. Lin, and C. C. Lu. 2017. "Effective Dynamic Dispatching Rule and Constructive Heuristic for Solving Single-Machine Scheduling Problems with a Common Due Window." *International Journal of Production Research* 55 (6): 1707–1719.

Zhang, Y., Y. Wang, J. Tang, and A. Lim. 2020. "Mitigating Overtime Risk in Tactical Surgical Scheduling." *Omega* 93: 102024.

Zhang, Z., and X. Xie. 2015. "Simulation-Based Optimization for Surgery Appointment Scheduling of Multiple Operating Rooms." *IIE Transactions* 47 (9): 998–1012.

Zhao, Z., and X. Li. 2014. "Scheduling Elective Surgeries with Sequence-Dependent Setup Times to Multiple Operating Rooms Using Constraint Programming." *Operations Research for Health Care* 3 (3): 160–167.

Zhu, S., W. Fan, S. Yang, J. Pei, and P. M. Pardalos. 2019. "Operating Room Planning and Surgical Case Scheduling: a Review of Literature." *Journal of Combinatorial Optimization* 37 (3): 757–805.

Appendix A. Notations

Table A1. Summary of Notations

	Basic Notations
I	number of operating rooms
i	index of operating room, $i = 1, ..., I$
J	number of surgeries
j	index of surgery, $j = 1, ..., J$
K	number of surgeons
k	index of surgeon, $k = 1, ..., K$
	Notations about Constraints
$\mathbb{M}_j$	a set of operating rooms which can process surgery j
M_j	number operating rooms in $\mathbb{M}_j$
$\mathcal{A}$	the *eligibility matrix* of $\mathbb{M}_j$
R_j	the surgeon dedicated to surgery j, $R_j = 1, ..., K$
	Notations in MILP Formulation
$x_{i,j}$	whether to schedule surgery j in operating room i
$y_{j,j'}$	whether surgery j is before surgery j'
s_j	the staring time of surgery j
C_{max}	the makespan
B	a big number
	Notations in Adaptive Composite Dispatching Method
t	notation for the time
$I_j(i,t)$	the ranking index for surgery j in operating room i at time t
$\mathbb{U}(i,t)$	the set of surgeries which can be processed in operating room i and have not been dispatched
$\mathcal{P}$	the total processing time of all surgeries
$\overline{M}(i,t)$	the average of M_j's of the surgeries that can be processed in operating room i and have not yet been dispatched at time t
$\mathcal{P}(R_j,t)$	the total processing time of un-dispatched surgeries of surgeon R_j at time t
$\overline{\mathcal{P}}(i,t)$	the average of $\mathcal{P}(R_j,t)$ for all surgeries which can be processed in operating room i and have not been dispatched at time t
σ_1	scaling parameter
σ_2	scaling parameter
φ	surgery flexibility
ψ	operating room flexibility
ρ	process flexibility index
ϕ	surgeon workload balance
	Notations for Sequential Uniform Design
t	the current iteration
$D_i^{(t)}$	the domain for σ_i in iteration t, $D_i^{(t)} = [a_i^{(t)}, b_i^{(t)}]$ and $c_i^{(t)} = (b_i^{(t)} - a_i^{(t)})/2$
q	the number of levels for each parameter
ϵ	a number which is small enough
β	a predefined contraction ratio
$n^{(t)}$	the number of pairs in iteration t
	Notations for Modified Simulated Algorithm
ST	maximum stages
IT	maximum iterations at each stage
p_a	acceptance probability
s	the current schedule
$V(s)$	makespan of schedule s
$\tilde{r}$	a random number

Appendix B. Sequential Uniform Design

In this section, we illustrate how to employ the so-called "Sequential Uniform Design" method (Fang and Lin 2003) to search for the best pair of σ_1 and σ_2 for each instance, which are used in ranking index to generate near optimal schedules.

The uniform experimental design is one kind of space filling designs for experiments when the underlying model is unknown (Fang and Lin 2003). It seeks the design points to be uniformly scattered on the experimental domain. It is able to explore relationships between the response and the factors with a reasonable number of runs and is shown to be robust to the underlying model specifications. For practical ease of use, most uniform designs have been specified and tabulated, which are available on the Internet[2]. One can also build her/his own uniform design table and there are numerous methods to construct it, for instance, the *Good Lattice Point* method (see Fang and Lin 2003, for details).

Sequential uniform design, also called sequential number-theoretic optimization method, is developed by Fang and Wang (1993). They use the number-theoretic method to generate uniformly scattered points and suggest a sequential algorithm for optimization. For our problem, we apply the sequential uniform design to find the best pair of σ_1 and σ_2 for a given instance with the same accuracy but less number of experiments than that using the normal uniform design.

Next, we generalize the algorithm of the sequential uniform design and combine it with the framework of sequential number-theoretic optimization algorithm (Zhang et al. 1997) to obtain σ_1 and σ_2.

In Algorithms 3, B determines the initial searching range for the scaling parameters. We suggest setting a large B to avoid missing possible best values. ϵ is to stop searching when the searching range is smaller than it. How small it should be depends on how sensitive the generated schedule is to the scaling parameters. q and $n^{(t)}$ determines how many possible pairs of scaling parameters to check in each iteration. The algorithm is more accurate when the design is larger, and it takes less time when the design is smaller. Following the suggestions by Fang and Wang (1993), for the sake of both effectiveness and efficiency, we choose a big uniform design firstly to construct the possible range of optimal combination more precisely, and then we take a smaller uniform design to locate the optimal combination more quickly. β is the step size

[2]For example, https://www.math.hkbu.edu.hk/UniformDesign/

Algorithm 3 Sequential Uniform Design

```
1: Set $t = 1$, $D_i^{(1)} = [0, B]$, $a_i^{(1)} = 0$, and $b_i^{(1)} = B$, where $B$ is a big number;
   $D_i^{(t)}$ is the domain that $\sigma_i$ may possibly take in iteration $t$. Choose a proper $q$ as the number of levels for each parameter;
   Choose a proper $\epsilon$ which is small enough; Choose a proper $\beta$ as a predefined contraction ratio to scale down the domain for each parameter.
2: Choose a proper uniform design table, $U_{n^{(t)}}(q^2)$, where "2" stands for the number of parameters, and $n^{(t)}$ is the number of pairs in iteration $t$.
   By applying the specific uniform design table, we can obtain $n^{(t)}$ pairs of $\sigma_1$ and $\sigma_2$ uniformly scattered on $D_i^{(t)} = [a_i^{(t)}, b_i^{(t)}]$.
3: Calculate the objective value with each pair of $\sigma_1$ and $\sigma_2$ by applying the corresponding ranking index;
   Set the pair that generates the minimum objective value as the new center of $\sigma_1$ and $\sigma_2$, denoted by $(\sigma_1^{(t)}, \sigma_2^{(t)})$.
4: Set $c_i^{(t)} = (b_i^{(t)} - a_i^{(t)})/2$. If $\max c_i^{(t)} < \epsilon$, then $D_i^{(t)}$ is small enough; the pair $(\sigma_1^{(t)}, \sigma_2^{(t)})$ is acceptable and STOP.
   Otherwise, go to step 5.
5: Set $D_i^{(t+1)} = [a_i^{(t+1)}, b_i^{(t+1)}]$ as follows.
       $a_i^{(t+1)} = \max(\sigma_i^{(t)} - \beta c_i^{(t)}, 0),$    (B1)
       $b_i^{(t+1)} = \min(\sigma_i^{(t)} + \beta c_i^{(t)}, B),$    (B2)
   Set $t = t + 1$. Go to step 2.
```

of shrinking the searching range. The algorithm converges faster when it is smaller. According to Fang and Wang (1993), 0.5 is a common choice. The details about the efficiency and robustness of the design, as well as the user's guide, can be found in Fang (1994). Sequential Uniform Design can be used to search for scaling parameters for any kind of composite dispatching rule. For a specific problem, we suggest constructing a pre-training data set to first locate the proper hyper-parameters and using a validation data set to tune these hyper-parameters.

Appendix References

Fang, K. T. 1994. "Uniform Design and Uniform Design Table." *Science Press, Beijing, China* .

Fang, K. T., and D. K. J. Lin. 2003. "Ch. 4. Uniform Experimental Designs and Their Applications in Industry." *Handbook of Statistics* 22: 131–170.

Fang, K. T., and Y. Wang. 1993. *Number-Theoretic Methods in Statistics*. Vol. 51. CRC Press.

Zhang, L., Y. Z. Liang, R. Q. Yu, and K. T. Fang. 1997. "Sequential Number-Theoretic Optimization (SNTO) Method Applied to Chemical Quantitative Analysis." *Journal of Chemometrics* 11 (3): 267–281.