Submitted to *Manufacturing & Service Operations Management*
manuscript 16-464

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

Operating-Room Staffing and Scheduling

(Authors' names blinded for peer review)

Problem definition: We consider two problems faced by an Operating-Room (OR) manager: (1) how many baseline (core) staff to hire for OR suites, and (2) how to schedule surgery requests that arrive one by one. The OR manager has access to historical case count and case length data. He or she needs to balance the fixed cost of baseline staff and variable cost of overtime, while satisfying surgeons' preferences.
Academic/practical relevance: ORs are costly to operate and generate about 70% of hospitals' revenues from surgical operations and subsequent hospitalizations. Because hospitals are increasingly under pressure to reduce costs, it is important to make staffing and scheduling decisions in an optimal manner. Also, hospitals need to leverage data when developing algorithmic solutions, and model tradeoffs between staffing costs and surgeons' preferences. We present a methodology for doing so, and test it on real data from a hospital.
Methodology: We propose a new criterion called the *robust competitive ratio* for designing online algorithms. Using this criterion and a Robust Optimization (RO) approach to model the uncertainty in case mix and case lengths, we develop tractable optimization formulations to solve the staffing and scheduling problems.
Results: For the staffing problem, we show that algorithms belonging to the class of interval classification algorithms achieve the best robust competitive ratio, and develop a tractable approach for calculating the optimal parameters of our proposed algorithm. For the scheduling phase, which occurs one or two days before each surgery day, we demonstrate how a robust optimization framework may be used to find implementable schedules while taking into account surgeons' preferences such as back-to-back and same-OR scheduling of cases. We also perform numerical experiments with real and synthetic data, which show that our approach can significantly reduce total staffing cost.
Managerial implications: We present algorithms that are easy to implement in practice and tractable to compute. These algorithms also allow the OR manager to specify the size of the uncertainty set and to control overtime costs while meeting surgeons' preferences.

Key words: Operating rooms staffing, Operating Room Scheduling, Robust Optimization

1. Introduction

Operating rooms (ORs) are costly to operate and generate about 70% of hospitals' revenues from surgical operations and subsequent hospitalizations (Jackson 2002). ORs are staffed by surgeons and anesthesiologists who may not be salaried, and teams of salaried staff consisting of nurse anesthetists, OR technicians, surgical technicians, scrub and circulating nurses, and first-assistant nurses. Because a significant portion of the estimated $15-20 per-minute cost of a fully-staffed OR can be attributed to staff salaries (Macario 2010), OR managers are often interested in establishing an optimal baseline (core) staffing level. Baseline staffing also impacts the cost of contingent staff (overtime, float pool, on-call, and contract workers) that hospitals use to meet realized excess demand for staffed-OR time. In order to determine an optimal baseline staffing level the hospital must account for case scheduling practices, which impact the utilization of staffed ORs. Therefore, the objective of this paper is to present a data-driven methodology that determines (1) the baseline staffing level, and (2) the surgical case schedules. We accomplish the staffing and scheduling tasks in two steps, which we call Phase I and Phase II of our approach. Before describing these phases, we explain typical staffing and surgical case scheduling practices at community hospitals.

1.1. Institutional Background

Baseline staff are typically hired under a long-term contract. Hospitals develop biweekly (or longer) work schedules for their baseline staff, usually several weeks in advance. Staff schedules determine the hospital's ability to open a certain number of ORs on each future day. Note that a hospital may open a different number of ORs each day of the week, although their baseline staff remains constant, by adjusting work patterns of baseline staff, utilizing contingent staff, or asking core staff to work overtime.

Hospitals use either *open* or *block* scheduling, with many US hospitals opting for block scheduling. A description of these practices can be found in Hopp and Lovejoy (2013, Chapter 4). In a block schedule, surgeons are guaranteed blocks of specific OR times on specific days of the week, whereas in open scheduling there are no guaranteed allocations and cases are booked on a first-come-first-served basis. Many US hospitals assign a portion of the available OR time as blocks, and keep the rest open for surgeons without block privileges. Whereas many papers in the operations management (OM) literature consider either pure block or pure open scheduling, we model a variant of block scheduling protocol that is used in many community hospitals. In this scheme, surgeons or surgical services are guaranteed blocks of time, but not necessarily in a particular OR. Also, the hospital exercises control over booking cases into blocks, and blocks into ORs. There are two reasons why we consider this approach. First, many hospitals do not transfer complete control of booking cases to surgeons. Examples where hospitals maintain control over scheduling blocks can be found in Benchoff et al. (2017) and Denton et al. (2010). The former describes scheduling practices at

Kaiser Permanente and the latter at Mayo Clinic. We also present a specific example in the next paragraph. Second, we show in this paper that our approach results in significant cost savings while honoring block commitments and surgeons' preferences. Thus, our approach may be viewed as an alternate prescriptive approach.

Our example comes from a community hospital that shared 18 months of surgical scheduling data with us. Using this data, we find that nearly 22.7% of all scheduled cases (2441 out of 10,731) comprised of instances in which a block surgeon operated in at least one other room on the same day, providing evidence that surgeons do operate in multiple rooms. We also found that all same-surgeon cases that were placed in one OR were scheduled back to back. The community hospital kept track of block usage and honored block commitments by guaranteeing placement of a case if the scheduled case length fitted in the remaining block time of that surgeon. However, surgeons did not own time in a particular OR for their block. A few days before each surgery day (typically 2-3 days before), all deferrable cases were known and the hospital re-optimized surgery schedules to achieve increased efficiency. The hospital used planned case lengths for scheduling purposes, which were based on an average of realized case lengths of recent similar cases. Our two step approach works in a similar way.

1.2. Our Approach

In Phase I of our approach, we use an interval classification algorithm to place surgical cases into virtual ORs one at a time. Its purpose is to find the long-term minimum number of ORs needed to accommodate block surgeons' cases that will be booked in an online fashion. We prove that the search for an algorithm that yields the smallest competitive ratio may be restricted to the set of interval classification algorithms, and find optimal interval breakpoints for the unknown daily case mix that lies in an uncertainty set. This implies that our algorithm is one of the best among algorithms that achieve the highest packing efficiency of block surgeons' cases in the worst case. Note that the hospital must schedule a case if the case fits in the remaining block time of a surgeon. We use data to estimate the number of ORs required for each day within a range of days and then determine the constant cost-minimizing core-staffing level across all days. Staffed OR demand that is not met by core staff is satisfied with the help of overtime, which cost more per unit of OR time. We utilize the trade-off between baseline (fixed) and overtime (variable) costs to determine the optimal baseline staffing level.

In an actual implementation, Phase I will be solved infrequently – say, once every year. It will determine the baseline staffing level. On a daily basis, cases for future days will be scheduled into virtual ORs as booking requests arrive such that each surgeon's requests are honored so long as they fit in his or her block time. This online scheduling of cases may be done by any convenient approach, including the approach we propose for finding baseline staffing. Then, one or two days before each surgery day, a detailed and implementable schedule will generated using Phase II .

In Phase II , we model physicians' preferences for back-to-back scheduling, placement of same-surgeon cases in the same OR, and penalize delays as well as idle time of surgeons. Phase II coincides with the common practice of reworking surgery schedule one or two days before each operating day to find a better packing of cases into blocks, and of blocks into ORs. At this stage, all block surgeons' cases are known and no previously-accepted case is denied, but blocks may be shifted to find a more efficient fit.

After a back-to-back schedule is created, surgeries may be sequenced according to the surgeon's preference without affecting the anticipated delays for the surgeon who holds the next block. This is possible because schedules are optimized a few days before each surgery day and patients are typically asked to arrive well in advance of the time when their surgeries are scheduled to start. Sometimes, it is necessary to schedule a surgeon's cases in different ORs. This occurs, for example, when some surgeries require special equipment that is available only in a few high-demand ORs (e.g. ORs with robotic surgery equipment). Phase II of our approach accommodates such situations as well.

In both phases, the uncertainty is modeled using the Robust Optimization (RO) approach through the use of uncertainty sets constructed from past data. In particular, the unknown surgery case mix in Phase I and the unknown case lengths in Phase II are modeled by appropriate uncertainty sets in Sections 3 and 4, respectively. These uncertainty sets are characterized by a parameter Γ which allows the OR manager to express her confidence in the data. In an alternative interpretation, the parameter Γ may be viewed as a budget of uncertainty or as a protection level as discussed in Bertsimas and Sim (2004). We also discuss the implications of the choice of Γ in Sections 3.1 and 6.3.

1.3. Contribution

Determining optimal baseline staffing for ORs is a hard problem because cases are booked one at a time when daily case mix and case lengths are unknown. Fitting surgical cases into

ORs is an online variant of the bin-packing problem, which is a known hard problem. In fact, nearly all previous papers in the OM literature deal with the offline problem in which a day's case composition is known – see Section 2 for further comparisons with the literature.

We make technical contributions in both phases of our approach. In Phase I we divide surgery lengths into different buckets (called intervals) and assign surgeries in the same interval into the same OR until the OR's total capacity is filled. By searching over all interval lengths in what is referred to as an interval classification algorithm, we find the best interval breakpoints for case-mix instances belonging to an uncertainty set. In particular, we prove that there exists an interval-classification based algorithm that minimizes the competitive ratio (CR), and present a tractable formulation that allows an OR manager to calculate the optimal interval breakpoints. In Phase II , we present a formulation for scheduling surgical cases that accommodates surgeons' preferences. Phase II is solved after knowing the case mix but before knowing the case lengths. The case lengths belong to a general polyhedral uncertainty set, whereas previous similar works consider interval sets. In this sense, Phase II is a generalization of earlier similar approaches. We also show that the Phase-II optimization problem is computationally tractable.

From a practitioner's perspective, this paper contributes by presenting a methodology for solving the staffing and scheduling problems in a common framework, modeling online placement of cases for staffing calculations, modeling surgeons' preferences for implementable case schedules, and demonstrating how this approach can be applied when the hospital has access to historical data. We also perform a variety of robustness checks to confirm that the predicted cost savings remain mostly intact when certain underlying assumptions are changed. Another key feature of our approach is that we utilize a robust optimization methodology and construct uncertainty sets from historical data. In this way, we ensure that our approach utilizes data without over fitting an assumed model of uncertainty to the historical data. As we demonstrate later, this allows us to gain the benefits of using data when the future realizations are similar to the historical data, while also being robust when future realizations deviate from that data.

2. Literature Review

This paper is related to two separate bodies of literature. The first contains papers on OR capacity management and the second on scheduling of surgical cases. The first group includes papers on bin-packing because from the mathematical modeling perspective bins are ORs, and jobs (also called items or packets) are surgical cases. Jobs (surgery-case booking requests) arrive one at a time giving rise to an online bin-packing problem. The objective is to pack an unknown set of jobs with different sizes in as few bins as possible. Utilizing this perspective, we review the two bodies of literature in separate subsections.

2.1. OR Capacity Management

The key question considered in this group of papers is "what is an optimal number of ORs?" Typically, no distinction is made between the physical number of ORs and the number of staffed ORs – the latter being a subset of the former. Moreover, in all OM papers that deal with OR capacity issues, the distribution of surgical-case demand (i.e. the daily number of cases and their case lengths) is assumed to be known. Put differently, previous works consider the stochastic version of the capacity-choice problem with complete distributional information, whereas we consider the online version.

Goldman and Knappenberger (1968) model the problem of determining an optimal number of ORs via a simulation model. More recently, Lovejoy and Li (2002) develop a queueing-theoretic model in which the OR manager decides the daily number of surgical cases to schedule per OR, and the probability that a case will be started on time. Given these two parameters, the amount of time that each OR needs to be open daily is determined optimally. The trade-off is between cost of capacity and cost of longer waits for patients. In contrast, we find an efficient level of staffing, after taking into account the inefficiencies introduced by the combination of finite work shifts and discrete case lengths, and online scheduling of cases. Other papers in the OM literature consider the problem of optimal nurse staffing, e.g. Yankovic and Green (2011) and Véricourt and Jennings (2011). These works use queueing models, ignoring finite shift lengths. In the context of baseline staffing for ORs, shift-length constraints can be a source of significant efficiency loss when placing surgical cases in ORs. Therefore, such approaches are not directly applicable to the problem of determining baseline staffing for ORs.

Online Bin Packing

As mentioned in the Introduction section, the problem of placing cases into ORs is an instance of the online bin-packing problem, which is well-studied in the computer science literature. Many algorithms have been proposed in this literature and their worst-case performances have been characterized. For example, the *Next Fit* algorithm, was proposed by Johnson (1973) with a competitive ratio of 2. Subsequently, it was shown by Johnson et al. (1974) that the *First Fit* algorithm had a competitive ratio of 1.7, which was improved to 5/3 by Yao (1980) who proposed the *Revised First Fit* algorithm. The best known competitive-ratio bound is by Seiden (2002) who showed that the *Harmonic++* algorithm had a performance ratio of at most 1.58889. A survey of the literature on online bin-packing algorithms is available in Coffman Jr. et al. (2013).

The algorithms mentioned above do not utilize historical data. We take a different approach, because some data is usually available. We constrain the set of possible job sequences for which the algorithm must guarantee performance to lie within an uncertainty set characterized by the historical data and a parameter Γ that determines its size. Therefore, unlike previous works, our approach does not yield a numerical CR bound applicable to all problem instances. Instead, we calculate a set of Γ-dependent and data-specific optimal interval breakpoints such that no other online algorithm has a lower asymptotic CR than our algorithm for the same data set and Γ.

2.2. Scheduling Surgical Cases

Surgical-case scheduling is a well-studied problem in the OM literature – see recent surveys in Guerriero and Guido (2011) and May et al. (2011). We classify these papers based on two aspects: (1) knowledge of case-length distributions, and (2) knowledge of the entire sequence of cases (online or offline). In Table 1, we present a classification of a subset of papers.

Table 1 A Classification of the Literature on Surgical-Case Scheduling

Distributional Information	Offline	Online
Known	Kong et al. (2013) Denton and Gupta (2003) Batun et al. (2011)	Gerchak et al. (1996) Dexter et al. (1999a) Dexter et al. (1999b)
Unknown	Mittal et al. (2014) (single OR) Denton et al. (2010)(multiple ORs)	**This paper** (multiple ORs)

In Phase I , we schedule cases in an online fashion, at which point case-mix and case-length distributions are unknown. In contrast, no previous paper considers both these aspects – see Table 1 – and as such those approaches cannot be directly applied to our setting.

The Phase II of our approach is similar to Li et al. (2016)'s OR schedule re-optimization problem. Li et al. improve an existing solution that already satisfies surgeons' preferences and all practical constraints. They are not concerned with how the initial solution is obtained. In contrast, we use an interval-based classification of planned surgery lengths to construct an initial solution that may not satisfy surgeons' preferences. The Phase II in our approach generates a feasible schedule while accounting for the possible discrepancy between planned and actual case lengths via a robust optimization framework. We demonstrate that our problem formulation can be solved efficiently with the help of commercial solvers. Li et al. (2016), in contrast, focus on modeling two shift lengths and obtaining bounds that are used in a customized branch-and-bound algorithm.

3. Phase I : Baseline Staffing

In this section we define the Robust Optimization (RO) problem considered in Phase I. We show how to determine the uncertainty set from historical data, and propose a new performance criterion called the Robust Competitive Ratio. This new performance criterion generalizes the standard competitive ratio (CR) which is defined for problems without any uncertainty set information. We propose an online algorithm for placing surgical cases into ORs and prove that no other algorithm can achieve a better competitive ratio so long as the case-mix uncertainty belongs to an uncertainty set, which is determined from the data.

3.1. Uncertainty Set Characterization

We divide possible case lengths into a maximum of N intervals and count the fraction of case lengths in each interval. We let $\mathcal{U}(\Gamma)$ to be the uncertainty set on the fraction of cases in each interval, where Γ determines the size of the set $\mathcal{U}$. We focus in this section on explaining the composition of the set $\mathcal{U}(\Gamma)$, and providing some intuition behind how OR managers may select Γ. The problem primitives are (i) arbitrary sequences L, (ii) normalized scheduled case lengths $(p_1, \cdots, p_m)$, where $p_i \in (0,1]$ for each L of size m, and (iii) ORs of capacity 1. In addition, some parameters are obtained from the historical data. For example, in our numerical experiments, we use data from a hospital that opens ORs for 600 minutes, i.e., 1 unit of time = 600 minutes and the minimum planned surgery duration is 15 minutes. Therefore, if we were to select interval breakpoints from the set of Harmonic breakpoints, then at most $N = 40$ breakpoints are needed. The breakpoints are such that $1 = t_1 > t_2, \cdots, t_N > t_{N+1}$, the ith breakpoint is at $t_i = 1/i$, and t_{N+1} either equals 0 or an arbitrary $\epsilon > 0$. For example, for our data, any positive value of ϵ smaller than 0.025 $(= 15/600)$ will suffice. Throughout this paper we set $t_{N+1} = 0$, but assume that there is a non-zero minimum length of any surgery request. An asterisk is affixed to decision variables to denote their optimal values. Table 2 contains a summary of the key notation used in our approach.

Table 2 Notation

Problem parameters:	
L	a set of surgeries ordered by arrival times, with ties broken arbitrarily
$\mathcal{S}$	set of all surgeons
$\mathcal{J}_s$	set of surgeries of surgeon s, $\forall s \in \mathcal{S}$ (note that $\cup_{s\in\mathcal{S}}\mathcal{J}_s = L$)
$(p_j, \tilde{p}_j)$	scheduled and actual duration of the j^{th} surgery, respectively
Γ	a parameter chosen by the OR manager that controls the size of the uncertainty set
γ_k	cost of overtime in k^{th} OR
η_s	surgeon-s idle time penalty per unit time
δ_s	surgeon-s delay time penalty per unit time
Algorithm parameters:	
$\mathcal{A}$	an arbitrary, constant-space online bin-packing algorithm
N	the maximum number of intervals that may be chosen by the algorithm
$\{t_1, t_2, \dots, t_N\}$	potential interval breakpoints lying in (0,1]
f_{ij}, f_i	fraction of cases whose planned lengths lie in $(t_{i+1}, t_i]$ on day j or an arbitrary day
(μ_i, σ_i, d)	mean, standard deviation, and number of instances of f_i in the training data
$\mathcal{U}$	uncertainty set of unknown fractions of cases in each interval
I_i	interval $(\tau_{i+1}, \tau_i]$
z_i	number of cases whose planned lengths lie in I_i
$\hat{z}_i$	fraction of cases whose planned lengths lie in I_i
Decision Variables:	
K	number of intervals used by $\mathcal{A}$ ($K \le N$)
$\{\tau_1, \tau_2, \dots, \tau_K\}$	interval breakpoints used by $\mathcal{A}$, each τ_i is one of $\{t_1, t_2, \dots, t_N\}$
x_{ij}	a binary $(0,1)$ variable indicating if the i^{th} breakpoint $\tau_i = 1/j$

Let f_i denote the random fraction of cases in the ith interval on an arbitrary day of operations. Because f_i will vary from one day to the next, f_is are treated as random variables defined over the support $[0,1]$. The f_is can be seen as discrete approximations of the distribution of surgery lengths, and therefore, the uncertainty in surgery lengths can be modeled via the uncertainty in f_is. Next, we use the Generalized Central Limit Theorem (GCLT) to obtain the distribution of sums of f_i's. In particular, we make use of the following result of Nolan (1997).

Theorem 1. (Nolan 1997) *Let $M_1, M_2, \dots, M_v$ be a sequence of i.i.d. random variables, with mean μ and undefined variance. Then, $\left(\sum_{i=1}^{v} M_i - v\mu\right)/C_\alpha v^{1/\alpha} \sim M$, where M is a standard stable distribution with a tail coefficient $\alpha \in (1,2]$ and C_α is a normalizing constant.*

The GCLT belongs to a broad class of weak convergence theorems. These theorems express the fact that the limiting sums of many independent random variables tend to be distributed according to one of a small set of stable distributions. When the variance of the random variables is finite, the stable distribution is the normal distribution and the GCLT reduces to the CLT. The stable laws are more general than the CLT and allow us to construct uncertainty sets for heavy-tailed distributions as well.

Upon obtaining historical data, an OR manager who wishes to use our approach will divide the data into two parts. The Part-1 data will be used to construct the uncertainty sets and the Part-2 data will be used to determine the optimal baseline staffing level. We provide a complete example with data from a community hospital in Section 5. Suppose the Part-1 data consists of d days. The observed fraction f_{ij} on the jth day is assumed to be an independent random draw from the distribution f_i. If f_is follow a light-tailed distribution (i.e. $\alpha_i \approx 2$) with mean μ_i and standard deviation σ_i, then $C_{\alpha_i}=1$. In this case, the normalized quantities $(f_{ij}-\mu_i)/\sigma_i$, are random draws from a distribution with zero mean and unit standard deviation. Therefore, ~~Y_is are random variables with zero mean and unit standard deviation, i.e. $C_{\alpha_i}=1$. Given this and using the variable transformation $y_{ij}=(f_{ij}-\mu_i)/\sigma_i$,~~ $\mathcal{U}(\Gamma)$ can be ~~equivalently~~ formally written as follows:

$$\mathcal{U}(\Gamma)=\left\{(f_1,f_2,\dots,f_N) \quad \text{s.t.} \quad -\Gamma d^{1/\alpha_i} \le \sum_{j=1}^{d} \frac{f_{ij}-\mu_i}{\sigma_i} \le \Gamma d^{1/\alpha_i} \quad \forall i=1,\dots,N\right\}. \tag{1}$$

The value of Γ is chosen by the OR manager depending on her subjective belief on the validity of historical data to predict future scenarios. In particular, as motivated in Bertsimas and Sim (2004), parameter Γ acts as a protection level. A higher value of Γ allows us to choose a baseline staffing that accounts for higher deviations in the case-mix of future days, thus obtaining a more robust baseline staffing. In Section 6.3, we discuss the costs and benefits of using different values of Γ. Additionally, uncertainty sets that allow for correlation can be defined. Please send inquiries to the authors.

3.2. Robust Competitive Ratio

Let $\mathcal{A}$ be an arbitrary algorithm that requires $\mathcal{A}(L)$ ORs to place an arbitrary sequence of cases L. Suppose an optimal offline algorithm requires $OPT(L)=n$ ORs for the same sequence. The performance of $\mathcal{A}$ is measured by the asymptotic performance ratio, which is also known as the competitive ratio (CR) and given by

$$\text{CR}(\mathcal{A})=\limsup_{n\to\infty}\sup_{L}\left\{\frac{\mathcal{A}(L)}{OPT(L)}\,|OPT(L)=n.\right\}. \tag{2}$$

CR guarantees performance for *any* input for large n. We refer to it as the *worst-case* CR because sequences L are completely arbitrary.

Motivated by this, we define the Robust Competitive Ratio, or RCR, with respect to an uncertainty set $\mathcal{U}(\Gamma)$ as follows.

$$\mathrm{RCR}(\mathcal{A},\Gamma) = \limsup_{n\to\infty} \sup_{L\in\mathcal{U}(\Gamma)} \left\{ \frac{\mathcal{A}(L)}{OPT(L)} \,|OPT(L) = n \right\}. \tag{3}$$

Because $\mathcal{U}(\Gamma) \subseteq \mathcal{U}(\infty)$, where $\mathcal{U}(\infty)$ consists of completely arbitrary sequences L, it is straightforward to argue that $\lim_{\Gamma\to\infty} \mathrm{RCR}(\mathcal{A},\Gamma) = \mathrm{CR}(\mathcal{A})$ and $\mathrm{RCR}(\mathcal{A},\Gamma) \le \mathrm{CR}(\mathcal{A})\ \forall \mathcal{A}$.

Our goal is to design an algorithm $\mathcal{A}^*$ such that for each fixed Γ, we have

$$\mathrm{RCR}(\mathcal{A}^*,\Gamma) = \min_{\mathcal{A}} \mathrm{RCR}(\mathcal{A},\Gamma). \tag{4}$$

In (4), $\mathcal{A}$ is an arbitrary constant-space algorithm. The constant space use restriction is an important tractability requirement because we consider asymptotic performance ratio. It means that permissible algorithms may not open more than a finite number of ORs for placing surgical cases at any time during their execution.

3.3. Robust Interval Classification Algorithms

Seiden (2002) first introduced a class of algorithms known as *interval classification* (IC) algorithms. Our interval classification algorithm has a set of intervals defined by a $K-$partition $t_1 = 1 > t_2 > \ldots > t_K > t_{K+1} = 0$. Each interval I_j is given by $(t_{j+1}, t_j]$ for $j = 1, \ldots, K$. Note that these intervals are disjoint and that they cover (0, 1]. Given these intervals, all incoming requests are classified depending on which interval they belong to. In particular, a case of size s is said to be of type j if $s \in I_j$. After classification, we use a slightly modified *Next Fit* algorithm to pack cases, which works as follows: (1) each existing class of cases has at most one open bin; (2) if the current case fits into the open bin, then it is placed there, else the open bin is closed, a new bin is opened for that case class and the case is placed there. This is a linear-time (in number of jobs) online algorithm. Seiden (2002) showed that the online algorithm that achieved the best known CR, known as the *Harmonic++* algorithm, belonged to this class of algorithms.

Next, we build upon Theorem 3 of Lee and Lee (1985) that establishes the asymptotic optimality of an interval classification algorithm for the worst-case competitive-ratio criterion. We show that this result also holds if we consider the Robust Competitive Ratio criterion defined in Equation (3). That is, the class of IC algorithms contains an optimal algorithm according to the RCR criterion as well.

Theorem 2. *For every uncertainty set* $\mathcal{U}(\Gamma)$ *defined by parameter* Γ*, there exists an interval classification algorithm* $\tilde{\mathcal{A}}$ *defined by an appropriate set of intervals* $I_j = (t_{j+1}, t_j]$, $j = 1, \cdots, K$*, where* $t_1 = 1 > t_2 > \cdots > t_K > t_{K+1} = 0$*, such that* $RCR(\tilde{\mathcal{A}}, \Gamma) = RCR(\mathcal{A}^*, \Gamma)$.

We present the proof of Theorem 2 in the online supplement. In what follows, we explain the key idea behind this result with the help of an example. Suppose there are only three types of surgeries and each type has n_r requests. Suppose Type-1 surgeries are of length $0.55 - 2\epsilon$, Type-2 are of length 0.3+ϵ, and Type-3 are of length $0.15 + \epsilon$ each. Then, an optimal offline algorithm will require exactly n_r bins to pack these items. Now consider an optimal online algorithm that can maintain no more than m_o ORs open at any point in time. Let us denote it by $\mathcal{A}^*$. Suppose Type-1 surgeries arrive first, followed by Type-2 surgeries, and finally Type-3. $\mathcal{A}^*$ will open n_r ORs to assign Type-1 surgeries, because at most one Type-1 surgery can be assigned to each OR. Similarly, at most 3 Type-2 jobs can be accommodated in an open bin. Therefore, $\mathcal{A}^*$ will require at least $(n_r - 3m_o)/3$ additional bins to pack Type-2 items. The reason why we subtract $3m_o$ is that we are attempting to establish a lower bound on the number of bins that $\mathcal{A}^*$ will require. In the best case, the algorithm $\mathcal{A}^*$ could have had m_o open bins, each of which could accommodate at most 3 Type-2 cases.

Continuing in this fashion, $\mathcal{A}^*$ will require at least $(n_r - 6m_o)/6$ additional bins to pack Type-3 items.Therefore, the number of bins consumed by $\mathcal{A}^*$ must be at least $n_r(1 + 1/3 + 1/6) - 2m_o$. In the limit as $n_r \to \infty$, the competitive ratio of $\mathcal{A}^*$ goes to $(1 + 1/3 + 1/6) = 1.5$, which is also the CR achieved by an interval classification algorithm with the breakpoints $1 = t_1 > t_2 = 1/3 > t_3 = 1/6 > t_4 = 0$, when the number of surgeries in each interval is n_r. The proof of Theorem 2 generalizes this argument for an arbitrary number of surgery types and case counts.

3.4. Robust Optimization Formulation

For fixed K, let $t_1 = 1 \geq t_2 \geq \ldots \geq t_K \geq t_{K+1} = 0$ be the interval breakpoints. We first argue that it is sufficient to consider breakpoints that belong to the harmonic sequence. That is, it is sufficient to search for breakpoints t_i of the form $t_i = 1/j$, where $j \geq i$, $j \in \mathbb{N}$, and $\mathbb{N}$ is the set of natural numbers. To see this, consider a sequence L consisting of $|L|$ jobs. Suppose that L has $\hat{z}_i$ fraction of jobs lying in interval I_i, and let $\Phi_{\mathcal{A}}(L)$ denote the total cost of packing this sequence using the IC algorithm $\mathcal{A}$. Then, $\Phi_{\mathcal{A}}(L) \leq \Phi_{\mathcal{A}}(L^u)$, where L^u is a sequence with same number of jobs such that $\hat{z}_i$ fraction of jobs are of size t_i. The above inequality holds because for every sequence $L \in \mathcal{U}(\Gamma)$, the sequence L^u also belongs to the set $\mathcal{U}(\Gamma)$ and the maximum possible size of a job in interval I_i is t_i. Because we are interested in the worst-case performance, we seek to minimize $\Phi_{\mathcal{A}}(L^u)$ by choosing $\mathcal{A}$.

In what follows, we will use $\lfloor \cdot \rfloor$ to denote the integer floor and $\lceil \cdot \rceil$ to denote the integer ceiling of a number. With this notation, observe that up to $\lfloor 1/t_i \rfloor$ jobs of size t_i can be packed in a unit sized bin. Therefore, the total cost of the sequence L^u is given by $\Phi_{\mathcal{A}}(L^u) = \sum_{i=1}^{K} \lceil |L| \cdot \hat{z}_i / \lfloor 1/t_i \rfloor \rceil \leq \sum_{i=1}^{K} |L| \cdot \hat{z}_i / \lfloor 1/t_i \rfloor + K$. At this point, we perform a variable transformation given by $\tau_i = 1/\lfloor 1/t_i \rfloor$ and $\tau_{K+1} = 0$. Using $\hat{\boldsymbol{z}}$ to denote $(\hat{z}_1, \ldots, \hat{z}_K)$ and $\boldsymbol{\tau}$ to denote $(\tau_1, \ldots, \tau_K)$, we obtain the following *minimax* optimization problem

$$\min_{\boldsymbol{\tau}} \left\{ h(\boldsymbol{\tau}) = \left[\max_{\boldsymbol{z}} \sum_{i=1}^{K} |L| \cdot \hat{z}_i \tau_i \text{ s.t. } (\hat{z}_1, \ldots, \hat{z}_K) \in \mathcal{U}(\Gamma) \right] \right\}, \text{ s.t. } 1 = \tau_1 \geq \ldots \geq \tau_K \geq 0. \quad (5)$$

with $\{\tau_i\}_{i=2}^{K}$ being the key decision variables. In optimization problem (5), the constraint $(\hat{z}_1, \ldots, \hat{z}_K) \in \mathcal{U}(\Gamma)$ is a short form for the following constraints:

$$\hat{z}_k = \sum_{\{i \text{ s.t. } \tau_{k+1} < 1/i \leq \tau_k\}} f_i, \quad \forall k = 1, \ldots, K-1,$$
$$\hat{z}_K = \sum_{\{i \text{ s.t. } 0 < 1/i \leq \tau_K\}} f_i,$$
$$(f_1, \ldots, f_N) \in \mathcal{U}(\Gamma).$$

This formulation is difficult to solve because $\hat{\boldsymbol{z}}$, the vector of unknown ~~counts of surgeries~~ fraction of cases that lie in each interval depends on the interval breakpoints $\{\tau_i\}_{i=2}^{K}$. To overcome that problem, we prove that (5) is equivalent to a computationally tractable binary optimization problem when we use uncertainty sets of the form (1) discussed in Section 2. This result is presented in Theorem 3, where we use the binary decision variables x_{ij}s to determine the optimal harmonic breakpoints. In particular, $x_{ij} = 1$ if $\tau_i = 1/j$ (i.e. if τ_i is the j^{th} breakpoint), and 0 otherwise. Additional decision variables used in this formulation are (1) $\{y_{i,r,s}\}$ — binary variables similar to x_{ij}'s, and (2) a and b. The constraints and these extra variables in Eq. (6) are a result of linearization of the constraints in optimization problem (5). More details are provided in the online appendix in Section 2.

Given that the uncertainty set is a polyhedron, the inner maximization problem for optimizing $h(\boldsymbol{\tau})$ is a linear optimization problem and we appeal to strong duality to convert problem (5) into a single optimization problem in Theorem 3. Its proof is presented in the online supplement. Recall that d is the number of days in our data used to define the uncertainty set in Eq. (1).

THEOREM 3. *For every uncertainty set $\mathcal{U}(\Gamma)$ defined by parameter Γ, the optimal interval classification problem (5) is equivalent to the following optimization problem*

$$\min_{a,b,\boldsymbol{x},\boldsymbol{y}} \quad a\left(\Gamma d^{1/\alpha} + \sum_i \frac{\mu_i}{\sigma_i}\right) - b\left(-\Gamma d^{1/\alpha} + \sum_i \frac{\mu_i}{\sigma_i}\right) \tag{6a}$$

$$\text{s.t.} \quad \sum_{i=1}^{K} x_{ij} \le 1, \qquad \forall j = 1, \cdots, N, \tag{6b}$$

$$\sum_{j=1}^{N} x_{ij} = 1, \qquad \forall i = 1, \cdots, K \tag{6c}$$

$$x_{i,r} \le \sum_{s=1}^{r} x_{i-1,s}, \qquad \forall i \ge 2, r = 1, \ldots, N, \tag{6d}$$

$$\sum_{i=1}^{K} \sum_{s=w+1}^{N} \sum_{r=1}^{w} y_{i,r,s} \le \frac{a-b}{\sigma_w}, \qquad \forall w = 1, \cdots, N, \tag{6e}$$

$$y_{i,r,s} \le x_{i,r}, \qquad \forall i = 1, \ldots, K, \; r, s = 1, \ldots, N, \tag{6f}$$

$$y_{i,r,s} \le x_{i+1,s}, \qquad \forall i = 1, \ldots, K, \; r, s = 1, \ldots, N, \tag{6g}$$

$$y_{i,r,s} \ge x_{i,r} + x_{i+1,s} - 1, \qquad \forall i = 1, \ldots, K, \; r, s = 1, \ldots, N, \tag{6h}$$

$$\{x_{ij}\} \in \{0,1\}, \text{and } a, b \in \mathbb{R}. \tag{6i}$$

3.5. Performance Guarantee

In Theorem 4, we show how to obtain the Robust Competitive Ratio of our algorithm. A proof of Theorem 4 is presented in the online supplement.

THEOREM 4 **(Robust Competitive Ratio)**. *Suppose the ~~case counts~~ fraction of cases in each interval are modeled by an uncertainty set $\mathcal{U}(\Gamma)$, and let $\tau_1^* > \tau_2^* > \ldots > \tau_K^*$ be the optimal breakpoints. Then the performance of the Interval Classification scheduling algorithm characterized by $\{\tau_i^*\}_{i=1}^K$ is given by the solution to the following linear program*

$$\max \sum_{i=1}^{K} \hat{z}_i \tau_i^* \quad \text{s.t.} \quad \sum_{k=1}^{K} \hat{z}_k \tau_{k+1}^* \le 1, \quad (\hat{z}_1, \ldots, \hat{z}_k) \in \mathcal{U}(\Gamma). \tag{7}$$

Theorem 4 states that the competitive ratio of our algorithm may be obtained by solving a linear program after obtaining the optimal τ_i^*s for any value of Γ. The competitive ratio depends on the problem parameters and the uncertainty-set parameter Γ.

4. Phase II : Surgical Case Scheduling

In Phase II , we focus on generating an implementable surgery schedule, in which we capture physicians' preferences for back-to-back scheduling and placement of same-surgeon cases in the same OR, while accounting for potential delays as well as idle time of surgeons. This approach coincides with the common practice of reworking surgery schedule one or two days before each operating day to find a better packing of cases into blocks, and of blocks into ORs. At this stage, all block surgeons' cases are known and no previously-accepted case is denied, but blocks may be shifted to find a more efficient fit. Any add-on cases are scheduled, as common practice, in either the free slots available in an OR or as overtime.

Phase II consists of a *pre-processing step* and an *optimization step*. For each surgeon s, the pre-processing step identifies cases that are exempt from back-to-back scheduling requirement (typically, because they require a special OR). These cases are put aside, and all remaining cases are lumped together to create a single virtual case equal the sum of the case lengths of those cases. We then add this virtual case to $\mathcal{J}_s$, the set of surgeon-s cases, and remove the cases that were combined. With this operation, all cases of each surgeon that are placed in a block are guaranteed to be scheduled back-to-back. We use additional notation in this phase, which is presented in Table 3.

In the optimization step, we seek to obtain an implementable schedule while balancing three main objectives: (1) ensure no overlap of same-surgeon cases, (2) allow back-to-back case schedules; and (3) control the tradeoff between surgeon delays, idle times and overtime. Note that baseline staffing decisions have already been made and the hospital has access to the list of all surgery requests and their planned case lengths. In this phase, there is still uncertainty about actual case lengths, and we use the set $\mathcal{U}^p$ to model this uncertainty. We present a formulation based on an arbitrary $\mathcal{U}^p$ in Section 4.1 and three specific choices in the online supplement. Given $\mathcal{U}^p$, Phase II utilizes known case counts and planned surgery lengths to produce robust surgery schedules. We argue later in this section that the optimization problem is relatively easy to solve using commercial solvers. After implementing the surgery schedule generated by our approach, and observing the realized case lengths, we calculate the ex-post overtime, delay and idle time statistics over a set of test data in Section 5.

We next present the optimization-problem formulation where the number of staffed ORs is fixed, and a schedule is determined that permits at most u_k minutes of overtime in OR k. This can lead to an infeasible solution if u_k is not chosen carefully. If that happens, and maximum

Table 3 Additional Notation

Optimization Formulation Parameters:	
M_0	number of staffed ORs as chosen in Phase I
M	number of staffed ORs in the iterative step in Algorithm 1
N_m	number of surgeries scheduled in the m^{th} OR, for $m = 1, \dots, M$
BigM	a large and positive number. Choose BigM $\geq$ the maximum completion time of all cases
$\mathcal{U}^p$	the uncertainty set modeling the unknown surgery case lengths
u_k	the maximum overtime allowed for the k^{th} OR
Decision Variables:	
$\zeta_{i,s}$	delay experienced by surgeon s for the i^{th} surgery
$\kappa_{i,s}$	idle time experienced by surgeon s after i^{th} surgery
$T_{k,i}$	the planned start time of the i-th surgery of the k^{th} OR
E_k	the planned end time of the last surgery of the k^{th} OR
Internal Accounting Variables:	
$\xi_{s,j,k,i}$	binary variables that define the sequence of surgeries for each surgeon
$o_{h,j,k,i}$	binary variables that define the sequence of predecessors of surgeries
$\chi_{h,j,k,i}$	binary variables that define the sequence of successors of surgeries

overtime (u_k) is a hard constraint, it may be necessary to add a full OR in overtime and resolve the Phase II formulation We describe this iterative approach in Algorithm 1, where we iterate over different numbers of the staffed ORs and repeatedly solve the optimization formulation until either a feasible schedule is found or we conclude that u_k is too restrictive for the available physical capacity of ORs.

Algorithm 1 Iterative Phase II Algorithm

1. Initialize $M = M_0$
2. Solve the optimization problem in Section 4.1.
3. If the problem is feasible
   - Stop and report the final schedule.

   Otherwise
   - Increment M by 1, and repeat Steps 2 and 3.

Suppose Algorithm 1 determines that the hospital needs to open M ORs. Then, $(M - M_0)$ ORs are opened in overtime. The cost of opening them in overtime does not affect the objective function in (8). Therefore, it is added to the objective function in (8) at the end.

4.1. The Optimization Formulation

For each iteration in Algorithm 1, we fix the value of M and solve an optimization problem. In order to keep track of surgeon delays, idling and overtime use, we use three sets of binary accounting variables, which we describe next.

- $\xi_{s,j,k,i} = 1$ if the i-th surgery of the k^{th} OR is the j^{th} surgery of the s^{th} surgeon, and 0 otherwise.
- $o_{h,j,k,i} = 1$ if the j-th surgery of the h^{th} OR starts **before** the i-th surgery of the k^{th} OR, and 0 otherwise.
- $\chi_{h,j,k,i} = 1$ if the i-th surgery of the k^{th} OR is the **next** surgery after the j-th surgery of the h^{th} OR finishes, and 0 otherwise.

With the above notation in hand, we next describe the key components of the post-allocation optimization problem.

The Objective function:

The objective function minimizes a weighted combination of total worst-case overtime, delay and idle time costs, and is given by

$$\min_{\{E_k,\kappa_{i,s},\zeta_{i,s}\}} \quad \sum_{k=1}^{M} \gamma_k \cdot (E_k - 1)^+ + \sum_{s\in\mathcal{S}} \delta_s \sum_{i\in\mathcal{J}_s} \kappa_{i,s} + \sum_{s\in\mathcal{S}} \eta_s \sum_{i\in\mathcal{J}_s} \zeta_{i,s}. \tag{8}$$

In (8), variables E_k, $\kappa_{i,s}$ and $\zeta_{i,s}$ compute, respectively, the overtime, the idle time and delays.

Surgery sequencing constraints: The following set of constraints allow us to control the sequence of surgeries in each OR.

$$\sum_{h=1}^{M}\sum_{j=1}^{N_h}\sum_{k=1}^{M}\sum_{i=1}^{N_k} \chi_{h,j,k,i} \le \left(\sum_{k=1}^{M} N_k\right) - M \tag{9}$$

$$\sum_{h=1}^{M}\sum_{j=1}^{N_h} \chi_{h,j,k,i} \le 1 \qquad \forall k = 1,\dots,M \quad \forall i = 1,\dots,N_k \tag{10}$$

$$\sum_{k=1}^{M}\sum_{i=1}^{N_k} \chi_{h,j,k,i} \le 1 \qquad \forall h = 1,\dots,M \quad \forall j = 1,\dots,N_h \tag{11}$$

$$\sum_{j=1}^{i} \chi_{k,i,k,j} = 0 \qquad \forall k = 1,\dots,M \quad \forall i = 1,\dots,N_k \tag{12}$$

$$\sum_{j=i+2}^{N_k} \chi_{k,i,k,j} = 0 \qquad \forall k = 1,\dots,M \quad \forall i = 1,\dots,N_k - 2 \tag{13}$$

Constraint (9) ensures that all cases will be scheduled. It requires that every case, except the last, must have a successor. In particular, there are $\sum_{k=1}^{M} N_k$ cases, and therefore, $\sum_{k=1}^{M} N_k - M$ successors. Constraint (10) (respectively, (11)) guarantees that each case can have only one predecessor (successor); the inequality is necessary to deal with the first (the last) case. Constraints (12) and (13) use the global sequence of surgeries to ensure that surgeries are ordered appropriately within the same OR. In particular, these constraints reflect the property that for the i^{th} surgery in the k^{th} OR, no other surgery other than the $(i+1)^{th}$ surgery could be its successor, across all ORs. That is, the $(i+1)^{th}$ surgery need not be in the same OR, it might be in a different OR.

Delay control constraints: The following set of constraints allow us to control potential delays by choosing the start times using a robust optimization approach. We use the variables $T_{k,i}$ to determine the planned start time of surgery i of k^{th} OR.

$$\begin{aligned} &\forall h, k = 1, \dots, M: \\ &T_{k,i} \geq T_{h,j} + \tilde{p}_j - \text{BigM}\,(1 - \chi_{h,j,k,i}), \quad \forall \{\tilde{p}_j\} \in \mathcal{U}^p, \\ &\forall j = 1, \dots, N_h; i = 1, \dots, N_k; \end{aligned} \tag{14}$$

$$T_{k,1} = 0 \quad \forall k = 1, \dots, M \tag{15}$$

$$\zeta_{s,j} = \sum_{k,i} T_{k,i}\xi_{s,j+1,k,i} - \sum_{k,i} T_{k,i}\xi_{s,j,k,i} \qquad \forall j \in \mathcal{J}_s, \quad \forall s \in \mathcal{S}, \tag{16}$$

In particular, constraint (14) is used to update the start time of new surgeries based on the unknown duration of the surgeries using a robust optimization approach. If $\chi_{h,j,k,i} = 1$, which implies that the i-th surgery of the k^{th} OR is the next surgery after the completion of the j-th surgery of the h^{th} OR, then the start time of the i-th surgery of the k^{th} OR cannot be before the end of the j-th surgery of the h^{th} OR, which implies $T_{k,i} \geq T_{h,j} + \tilde{p}_{h,j}$. The constraint ensures this when $\chi_{h,j,k,i} = 1$, but when $\chi_{h,j,k,i} = 0$, the constraint becomes redundant. Moreover, constraint (14) is a robustness constraint and it is well known (see Bertsimas and Weismantel 2005) that this constraint can be reformulated into multiple linear constraints when $\mathcal{U}^p$ is a polyhedron.

An arbitrary polyhedral uncertainty set $\mathcal{U}^p = \{\tilde{\boldsymbol{p}} = (\tilde{p}_1, \tilde{p}_2, \dots) \mid \mathbb{B}\tilde{\boldsymbol{p}} \leq \boldsymbol{b}\}$ may be operationalized in different ways. For example, for a day with $\tilde{m}$ surgeries, we construct the *Stationary* uncertainty set, $\mathcal{U}^1$ specified as

$$\mathcal{U}^1 = \left\{ (\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_{\tilde{m}}) \;\middle|\; -\Gamma\tilde{m}^{1/\alpha} \leq \frac{\sum_{i=1,\dots,\tilde{m}} \tilde{p}_i - \tilde{m}\mu}{\sigma} \leq \Gamma\tilde{m}^{1/\alpha} \right\}. \tag{17}$$

For $\mathcal{U}^1$, the matrix $\mathbb{B}$ is a $2 \times \tilde{m}$ matrix and $\boldsymbol{b}$ is a two dimensional vector given by

$$\mathbb{B} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ -1 & -1 & -1 & \dots & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \tilde{m}\mu + \Gamma\sigma\tilde{m}^{1/\alpha} \\ -\tilde{m}\mu + \Gamma\sigma\tilde{m}^{1/\alpha} \end{bmatrix}.$$

Note that the value of $\tilde{m}$ will depend on each day and is available at the beginning of Phase II.

We next consider constraint (14) for a fixed set of values of (h,k,i,j). Then, constraint (14) is given by $T_{k,i} \geq T_{h,j} + \tilde{p}_j - \text{BigM}\,(1-\chi_{h,j,k,i}), \quad \forall \{\tilde{p}_j\} \in \mathcal{U}^p$, which is equivalent to the following constraint

$$T_{k,i} - T_{h,j} + \text{BigM}\,(1-\chi_{h,j,k,i}) \geq \max_{\{\tilde{p}_j\}\in\mathcal{U}^p} \tilde{p}_j. \tag{18}$$

Now applying strong duality to the RHS of the above constraint, we have $\max_{\{\tilde{p}_j\}\in\mathcal{U}^p} \tilde{p}_j = \max_{\mathbb{B}\tilde{\boldsymbol{p}}\leq \boldsymbol{b}} \tilde{p}_j = \min_{\tilde{\boldsymbol{q}}'\mathbb{B}\geq \boldsymbol{e}_j} \tilde{\boldsymbol{q}}'\boldsymbol{b}$, where $\boldsymbol{q}$ are the dual variables, and $\boldsymbol{e}_j$ is the vector with 1 in the jth position and 0s everywhere. By using techniques from (Bertsimas and Weismantel 2005) and strong duality, the constraint (18) is equivalent to the following set of linear constraints:

$$T_{k,i} - T_{h,j} + \text{BigM}\,(1-\chi_{h,j,k,i}) \geq \tilde{\boldsymbol{q}}'\boldsymbol{b}, \quad \tilde{\boldsymbol{q}}'\mathbb{B} \geq \boldsymbol{e}_j. \tag{19}$$

To summarize, constraint (14) is equivalent to the set of linear constraints (19), with $\boldsymbol{q}$ as additional intermediate decision variables.

Finally, constraint (15) initializes the start times of the first surgery in each OR to be zero, and constraint (16) calculates the delay of the k^{th} surgeon after that surgeon's j^{th} surgery using the start times $T_{k,j}$ variables.

Idle time control constraints: The following set of constraints allow us to calculate the idle time of each doctor across ORs. This is achieved by using the $\boldsymbol{o}$ variables which track the relative sequence of cases for different surgeons across ORs.

$$o_{k,j,k,i} = 1 \qquad \forall k=1,\dots,M \quad \forall i=1,\dots,N_k \quad \forall j=1,\dots,i-1 \tag{20}$$

$$o_{k,j,k,i} = 0 \qquad \forall k=1,\dots,M \quad \forall i=1,\dots,N_k \quad \forall j=i,\dots,N_h \tag{21}$$

$$o_{h,j,k,i} + o_{k,i,h,j} = 1 \qquad \forall k,h=1,\dots,M \quad \forall i=1,\dots,N_k \quad \forall j=i,\dots,N_k \tag{22}$$

$$\kappa_{s,j} = \sum_{k,i} T_{k,i}\xi_{s,j,k,i} - \sum_{k,i} T_{k,i}\xi_{s,j-1,k,i} \qquad \forall j\in\mathcal{J}_s, \quad \forall s\in\mathcal{S}, \tag{23}$$

Specifically, constraint (20) (respectively, (21)) guarantees that each case can have only one predecessor (successor); the constraint is necessary to deal with the first (the last) case. Constraints (22) use the global sequence of surgeries to ensure that surgeries are ordered appropriately. Finally, constraint (23) calculates the idle time of the s^{th} surgeon after the surgeon's j^{th} surgery using the start times $T_{k,i}$ variables.

Overtime control constraints: The following set of constraints allow us to calculate the overtime in each OR by calculating the end time of the last scheduled case in each OR.

$$E_k \geq T_{k,i} + \tilde{p}_i \quad \forall \{\tilde{p}_i\} \in \mathcal{U}^p, \quad \forall i = 1, \ldots, N_k, \quad \forall k = 1, \ldots, M, \tag{24}$$

$$E_k \leq 1 + u_k \quad \forall k = 1, \ldots, M, \tag{25}$$

Specifically, constraint (24) calculates the completion time of the final surgery in each OR. Constraint (25) imposes a constraint on the maximum overtime using the OR specific upper bound u_k.

Non-negativity and Binary constraints: Finally, we add the corresponding non-negativity and binary constraints:

$$\{o_{h,j,k,i}, \chi_{h,j,k,i}, \xi_{s,j,k,i}\} \in \{0,1\}, \{T_{k,i}\} \geq 0. \tag{26}$$

5. Computational Experiments With Real Data

In this section, we use real data from a midsize community hospital to demonstrate how our approach would be implemented in practice. We also perform what-if analyses to test the impact of the surgeon-delay cost and the single OR overtime limit.

5.1. Data

The data set consists of 18 months of surgical-case data from a community hospital with 11,227 scheduled cases. The hospital operated 10 ORs during that period that were open for 10 hours (600 minutes) per weekday. A different number of ORs were opened each day using baseline staff, and overtime was utilized to match available OR time with demand. In our data, there were 72 unique surgeon IDs, 683 unique primary procedure IDs, and 4 patient classes (Inpatient, Outpatient, Surgery Admits, and Emergency). The hospital used a moving average of recent similar cases to calculate scheduled case lengths. We deleted cases that were performed on weekends, or had a missing start-time or end-time stamp, or whose case lengths (either scheduled or actual) exceeded 600 minutes. The longer-than-600-minute cases were rare (only 17 out of 11,227 cases). They were dealt with on a case by case basis, and required special arrangements before they could be scheduled. They did not represent "regular" OR case load. After such pruning, we had 10,731 cases in our data set.

OR utilization is an important performance indicator. Therefore, we calculated total scheduled and actual minutes for which each OR was used each day, as well as the overtime minutes based on scheduled and actual case lengths. We included urgent and add-on cases when calculating overtime usage. The hospital recorded a scheduled case length associated with each case, some of which were not booked in advance. Then, we calculated the scheduled and actual utilization, conditional average overtime (when there is a non-zero overtime), and conditional average delays. These results are shown in Table 4.

Table 4 Basic Performance Statistics

OR	Avg. Utilization (%)		Avg. Overtime (mins)		Daily Avg. No.	Average Delay (mins)	
Number	Scheduled	Actual	Scheduled	Actual	of Delayed Cases	Mean	SD
1	74.5	72.0	22.4	34.2	3.2	11.3	38.6
2	51.8	59.3	9.1	21.2	1.1	73.2	61.3
3	55.4	55.3	2.4	3.7	2.8	29.7	55.4
4	49.1	45.4	1.3	3.7	2.9	19.4	48.7
5	55.1	51.9	0.9	1.1	3.1	17.3	55.8
6	56.6	55.4	1.5	1.9	3.2	32.1	48.3
7	54.0	57.2	2.9	2.9	1.8	39.3	71.5
8	70.8	62.3	12.8	7.1	2.9	7.2	59.4
9	67.0	50.0	1.1	0.0	2.6	9.8	39.4
10	78.5	65.5	39.3	16.3	1.7	7.6	78.6

Three statistics in Table 4 are noteworthy. First, different ORs operate differently. Some ORs have higher utilization than others. The difference comes from differences in case lengths of surgeries scheduled in these ORs. Second, the average overtime use in this hospital is low. The use of overtime depends on the case mix and operating policies. Low overtime usage may not be representative of US hospitals in general. Third, the standard deviation of delay is high relative to the mean. This suggests that while the vast majority of delays are small, delays can be sometimes large. This is in part because of the nature of surgical procedures. Unexpected complications may arise lengthening the procedure time. Therefore, some large delays are unavoidable.

5.2. Implementation Details

Implementation of our approach occurs in 5 steps presented below. Steps 1-3 belong to Phase I of our approach, while Steps 4-5 belong to Phase II of our approach.

- Step 1. We verified that the source distributions of the sequence $\left\{z_i^1, z_i^2, \ldots, z_i^d\right\}$ were independent distributions by using the *PowerLaw* package by Alstott et al. (2014). Using the same package, we also determined if the distributions are light-tailed or heavy-tailed. This package returns the value of the heavy-tail coefficient of the source distribution: a value of 2 for light-tailed distributions and a value in the interval $(1, 2)$ for heavy-tailed distributions. We then construct Phase-I uncertainty set $\mathcal{U}$ by using a value of Γ chosen by the OR manager.

- Step 2. We solve the optimization problem (6) and identified the optimal breakpoints τ_i^*'s for the interval classification algorithm.
- Step 3. We implemented our case allocation approach (with optimal breakpoints) to determine the baseline staffing.
- Step 4. We use Algorithm 1 to rework case-to-OR allocations to satisfy surgeons' preferences. This step would be performed one or two days before each surgery day after all cases were initially allocated to ORs.
- Step 5. Finally, we estimate total cost, overtime, delay, and idle time over real and synthetic data.

5.3. Performance Analysis

We implemented Steps 1-5 of Section 5.2. We found that the source distributions were light-tailed, implying that $\alpha_i = 2 \; \forall \; i$, would be appropriate for the hospital whose data we used. We choose $\Gamma = 2$ for both phases, $\gamma_k = \gamma = 1.5$, $\eta_s = \eta = 0$, and $\delta_s = \delta = 0$. Additionally in Phase II , we assumed that no OR could be assigned more than 15 minutes of overtime at the planning stage. This constraint resulted in some instances in which extra ORs (on top of what is possible with the baseline staffing level) were needed. Details are presented in Table 6. We also considered 30% of all cases to be exempt from back-to-back scheduling requirement. This number is slightly higher than the 22.7% of exempt cases we observed in the real data. In subsequent experiments, we also studied the impact of changing the percent of exempt cases. The exempt cases were placed in a randomly-selected OR. Upon solving Phase II , we obtained a set of surgery schedules for each day. Next, using these schedules and actual case lengths, we calculated ex-post overtime, delay, and idle times, as well as total cost over a 200-day test data set. Our results are presented in Figures 1 and 2.

In Figure 1, we overlay histograms of overtime use in actual data and the result of using our algorithm. The histograms are constructed with 20-minute wide bins and the vertical axis shows percent of days on which the overtime belonged to each bin. The actual overtime in the data is shown by solid (green) bars and the result from our approach is shown by unfilled bars. The average overtime used are also marked by vertical lines – actual at 93.7 minutes (standard deviation 72.9 minutes) and ours at 77.81 minutes (standard deviation 65.3 minutes). The p-value for the associated null hypothesis is 0.0515 indicating that the difference in means is statistically significant at 10% but not at 5%. These statistics are calculated after observing actual case lengths. Our approach results in many more days with

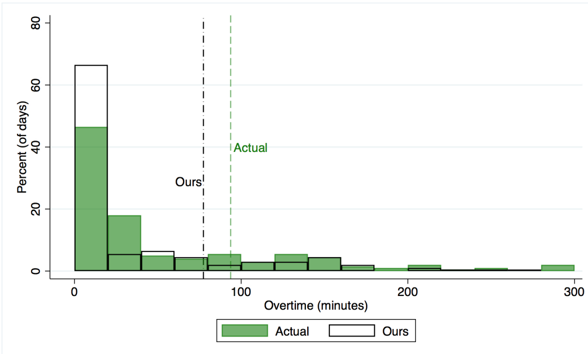

Figure 1 **Distribution of Overtime Used**

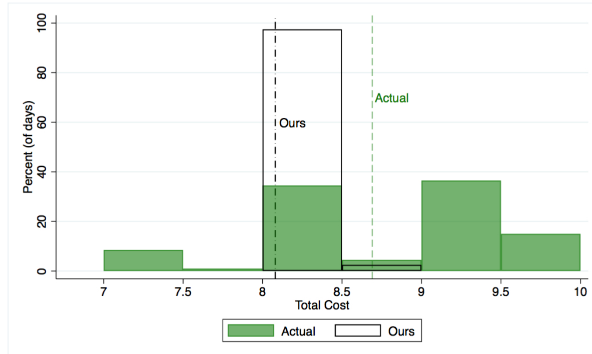

Figure 2 **Distribution of Cost Incurred**

zero overtime and in many cases, fewer days with non-zero overtime. However, when we compare the costs, we obtain a cost of 8.083 while the current approach leads to a cost of 8.69 with p-value ≈ 0, which indicates high statistical significance. It is noteworthy that our approach performs better than the current practice even though we do not vary how many ORs will be opened using baseline staff, whereas the hospital chooses different number of open ORs each day. Put differently, in our approach the cost of opening 7 ORs each day is sunk. Any additional OR usage is counted as overtime. In contrast, under the hospital's current approach, we count overtime usage only when an OR's actual closing time exceeds 600 minutes.

Next, in Figure 2, we compare the total cost distribution from the data (green bars) and our approach (unfilled bars). The width of these bars is chosen to be 0.25. Average total costs are shown by green dashed-dot line (actual) and black dotted line (ours). The average cost incurred by our algorithm is 8.1, which is much lower than 8.7 derived from the data. Figures 1 and 2 together show that our approach lowers cost while limiting overtime in any single OR to no more than 15 minutes. The cost savings come from a combination of online case placement in Phase I and optimal case scheduling in Phase II .

Because our approach packs surgical cases more efficiently and exempts some cases from back-to-back scheduling requirement, it is natural to ask whether the delays and idle times experienced by surgeons will increase. We calculated ex-post (after observing case lengths) delay and idle times from 200-day simulation using real data. The mean daily delay across all surgeons using our approach is 45.68 minutes (standard deviation 16.3, maximum 92.4), and the same statistic is 41.95 minutes (standard deviation 13.8, maximum 82.3) using the hospital's current schedule. The mean daily delays are statistically not different because their 95% confidence intervals overlap.~~The 95% confidence intervals over these two delay statistics overlap, suggesting that mean delays are statistically not different.~~ Similarly, the mean idle time across all surgeons using our approach is 68.1 minutes (standard deviation 19, maximum 126.7) and the same statistic is 58.9 (standard deviation 19, maximum 105.3) using the actual data.

~~Idleness is significantly affected by the percent of cases that are exempt from back-to-back scheduling requirement. So far, we had assumed that 30% of all cases were exempt. However, if we were to require that all cases be scheduled back-to-back, the surgeon idle times reduce to zero in our 200-day simulation. This comes at the cost of higher overtime and delay costs. With $\gamma_k = \gamma = 1.5, \delta_s = \delta = 0.3$, $u_k = \infty$, and 30% exempt cases, the average overtime and delays are 78.7 mins and 42.8 mins per day, respectively. When all cases must be back-to-back, these increase to 102.2 and 47.9 mins per day, respectively. That said, our approach has a better performance even with a small percentage of cases in the exempt category. For example, with 5% and 20% of cases in the exempt category, the average overtime and delays are, respectively, (89.3, 45.1) mins for the 5% instance, and (82.6,44.2) mins for the 20% instance, which are better than those in the current data where approximately 22.7% cases are in the exempt category.~~

We assumed that a constant baseline staffing is maintained across all days when our approach is used, but the hospital is able to adjust its daily staffing level to any level up to 10 ORs without incurring overtime charges. Also, we only counted the cost of the number of ORs actually opened by the hospital each day as it regular staffing cost. This flexibility lowers the hospital's total staffing cost shown in Figure 2 relative the costs incurred by our algorithm. In a real implementation, a hospital that uses our approach may be able to reduce costs further by utilizing a similar flexibility to adjust work schedules of baseline staff.

5.4. What-if Analyses

(I) Effect of Γ and Cost of delay

In this set of experiments, we highlight the relationship between baseline staffing and the cost of surgeon delay for different values of Γ. We fix the overtime cost $\gamma_k = \gamma = 1.5$ and the cost of ~~idel~~idle time $\eta_s = \eta = 0$, and vary $\delta_s = \delta$, the cost of delay. For each value of δ, we identify the range of values of Γ for which the optimal number of staffed ORs remains invariant. Results are shown in Table 5.

Delay Cost	Number of baseline staffed ORs			
δ	$\Gamma \in (0, 2.2]$	$\Gamma \in (2.2, 4.5]$	$\Gamma \in (4.5, 7.5]$	$\Gamma \in (7.5, \infty]$
0	7	8	9	10
0.1	8	8	9	10
0.2	8	8	9	10
0.3	9	9	9	10

Table 5 Number of Staffed ORs as a Function of Delay Penalty.

Table 5 shows that the optimal number of staffed ORs increases as the delay cost δ and the value of Γ goes up. In fact, opening all 10 ORs is optimal when the cost of delaying physicians is very high or when the value of Γ is high. Note that a delay cost of $\delta = 0.3$ translates to approximately $3,000-3,600 per day. This comes from the fact that cost of an OR shift of 600 minutes is normalized to 1 and that a minute of staffed OR costs about $15-20. When delay cost is very high, our algorithm assigns dedicated rooms to many surgeons.

(II) Single OR overtime limit

It is not ideal for any single OR (and its associated staff) to have a large amount of overtime. Many hospitals have arrangements with staff that up to 15 minutes of overtime will be permitted (see Dexter et al. 1999b). In the next set of experiments, we calculate performance statistics after imposing the constraint that the maximum overtime in any OR may not exceed either 15, 20, or 30 minutes (see Constraint 25). In Table 6, we report the total cost statistics (mean, standard deviation and 95th percentile), along with the fraction of times either one or two extra ORs are required for values of $u_k = 15$, 20, 30 minutes for all k. In these experiments, $\Gamma = 2$, $\gamma_k = \gamma = 1.5$, $\eta_s = \eta = 0$, and $\delta_s = \delta = 0$, and 30% of the cases are exempt from back-to-back scheduling requirement. As the overtime limit becomes more strict, the optimal baseline staffing as well as the total cost of ORs goes up, i.e. more baseline staff are hired at lower overall utilization. That being said, the hospital does not need more than 2 additional ORs on any given day.

Overtime	Baseline Staffing, Total cost (Regular + Overtime)	Fraction of times extra ORs are used		
u_k (mins)	(Mean, Stdev, 95th percentile)	0	1	2
15	8, (9.024, 0.19, 9.41)	36.8%	42.3%	20.9%
20	8, (8.82, 0.23, 9.28)	52.5%	35.3%	12.2%
30	8, (8.56, 0.26, 9.12)	73.8%	21.6%	4.6%

Table 6 Optimal Staffing with Permissible Overtime Choice of $\mathbf{u_k = 15, 20, 30}$ minutes.

6. Computational Experiments With Synthetic Data

In order to understand the performance of our approach under a variety of different data sets, we generated synthetic data in a format that is similar to the data from the community hospital. In the real data set, we have access to the total number of surgeries planned for a given day, the planned surgery lengths and the actual surgery lengths. We use the following *bootstrapping* method to generate similar synthetic data.

1. We created two sets $\mathcal{X}$ and $\mathcal{Y}$, where $\mathcal{X}$ contained all planned (or scheduled) case lengths in the historical data, and the set $\mathcal{Y}$ contained the total number of cases that were scheduled on each day in the data.
2. We computed the empirical distribution $\mathbb{F}^{e}_{\Delta}$ of the difference between the actual and scheduled case lengths, denoted by Δ, as a function of the surgery case type.
3. Next, we generated random samples as follows:
    - For each day, generate the random number of surgery requests $\tilde{n}$ by sampling with replacement from the set $\mathcal{Y}$.
    - We generated $\tilde{n}$ scheduled case lengths by sampling with replacement from set $\mathcal{X}$.
    - For actual case lengths, we generated random samples of Δ from either the empirical distribution $\mathbb{F}^{e}_{\Delta}$ or from arbitrary distributions $\mathbb{F}_{\Delta}$, and added Δ to the scheduled case lengths generated in the previous step. Note that Δ were sampled from the appropriate $\mathbb{F}^{e}_{\Delta}$ distributions that matched the scheduled case lengths and case types.

In all simulations we performed with synthetic data, we generated 1000 random instances, with each instance consisting of 100 days of surgery requests. Experiments were then performed and test statistics tabulated over these random instances. Note that in all these experiments, we set $\gamma_k = 1.5\ \forall k$, $\Gamma = 2$, $\delta_s = \eta_s = 0\ \forall s$ (i.e., zero delay and idle time costs), $u_k = \infty$, and allow 30% of all cases to be exempt from back-to-back scheduling requirement.

6.1. Performance Analysis

In this section, we demonstrate the performance of our approach relative to the Harmonic ++, the First Fit, and the Next Fit algorithms, denoted by H++, FF and NF, respectively. We begin by generating 1000 instances of sequences of surgeries, with each instance consisting of 100 days. Let L^i be the sequence of surgeries for 100 days in the ith instance, $\mathcal{A}$ be one of the online algorithms. We use each algorithm to separately compute the baseline staffing and then use the optimization problem in Section 4.1 to compute the total cost to serve each sequence L^i, denoted by $c^*_{\mathcal{A}}(L^i)$. Then, we calculate the ratio $r_{\mathcal{A}}(L^i) = c^*_{\mathcal{A}}(L^i)/c^*(L^i)$, where

$c^*(L^i)$ is the cost associated with our proposed algorithm. Thus, we have 1000 ratios across matched instances for three competing algorithms. We repeat these experiments with three sets of synthetic data in which case lengths were drawn from standard normal, exponential, and standard lognormal distributions, respectively. We compute summary statistics of these ratios and report them in Table 7.

Performance	Normal			Exponential			Lognormal		
Measure	H++	FF	NF	H++	FF	NF	H++	FF	NF
Average	1.41	1.51	1.57	1.33	1.37	1.42	1.44	1.48	1.55
5th Percentile	1.03	1.12	1.18	1.01	1.04	1.04	1.04	1.13	1.23
Min	0.90	0.98	1.14	0.90	0.94	0.90	0.93	0.99	1.12

Table 7 Performance of common algorithms relative to our algorithm.

Note that the average ratio ranges from 1.33 to 1.57. This means that competing ~~algorithm~~algorithms result in costs that are on average 33% to 57% higher than those obtained from our algorithm. The 5th percentile of the ratio is always greater than 1. That is, our algorithm produces a lower cost in 95% of test cases. Finally, the minimum ratio is smaller than 1. That is, there are some instances of case sequences for which our algorithm results in a larger cost. This is not surprising because our algorithm may not perform as well as others in all cases when job sequences are finite. Recall that our algorithm is designed to minimize RCR, which is the competitive ratio in the limit that the number of jobs in the sequence L goes to infinity (see Equation 2).

6.2. Robustness

We next demonstrate the benefit of using the robust optimization approach when the distribution of future uncertainty might differ from the historical data. To study this, we design the following experiments. We first assume that the future surgery lengths are distributed according to a distribution called the "true distribution". However, the historical data comes from a different distribution, which we call the "assumed distribution". The two distributions have the same mean and variance. Then, we investigate how the total cost of our approach compares with the stochastic offline optimal total cost, where the stochastic optimization (SO) approach uses the assumed distribution to obtain optimal schedules. Note that the RO approach uses the historical data to obtain optimal schedules. In each case, the total cost of a staffing plus scheduling solution is estimated by sampling daily case volume, case mix,

True Distribution	*Assumed Distribution* Beta	Gamma	Normal	Pareto	Triangular	Uniform
Beta	-0.121	0.559	0.162	0.292	0.789	0.305
Gamma	0.561	-0.098	0.252	0.821	0.227	0.253
Normal	0.468	0.676	-0.091	0.202	0.555	0.244
Pareto	0.195	0.386	0.731	-0.073	0.599	0.173
Triangular	0.553	0.588	0.178	0.171	-0.087	0.515
Uniform	0.566	0.25	0.537	0.42	0.673	-0.132

Table 8 **The *relative benefit* of using the RO approach — same mean and standard deviation.**

and case lengths from the true distribution. We calculate the *Relative Benefit* of the RO approach as follows:

$$\text{Relative Benefit} = \frac{\text{Cost of Stochastic Optimal Schedule} - \text{Cost of our Algorithm}}{\text{Cost of Stochastic Optimal Schedule}}, \quad (27)$$

The results are shown in Table 8. In each comparison in Table 8, we obtained a p-value ≈ 0 indicating high statistical significance. The relative benefit is negative only when the assumed and the true distributions are the same. In all other cases the relative benefit is positive, meaning that the RO approach outperforms the SO approach. In fact, when the assumed and the true distributions are different, the relative benefit is generally 0.2 (20%) or higher and may be as high as 0.82 (82%). The average relative benefit, not counting cases in which the assumed and the true distribution are identical, is 0.358 (35.8%).

There are other ways in which the true and the assumed distributions may be different. For example, the two distributions may have the same functional forms, but either different variances (for the same mean), or different means (for the same variances). The relative benefit of our approach with respect to such errors in estimates of distributional parameters is further explored in the online supplement (see Tables 10 and 11). In all these experiments, we find that our approach is superior to the SO approach when the assumed and the true distributions are different.

6.3. Choice of Γ

In this section, we use computational experiments to shine light on how an OR manager may select Γ. We perform and report results of three experiments. In the first two experiments, we assumed that the functional forms of the historical and the future distribution of Δ are the same. The former is called the *assumed* and the latter, the *true* distribution. In our first experiment, the mean value of Δ in the assumed distribution is some μ and the standard deviation is $\sigma = 1$. The true distribution's standard deviation equals 1, but its mean could be either $\mu/4$, or $\mu/2$, or μ, or 2μ or 3μ. We calculated the relative benefit of using the

RO approach over the SO approach in each case for 5 different values of Γ. In the second set of such experiments, the assumed and true distributions have the same mean $\mu = 1$, and that the standard deviation of the assumed distribution Δ is $\sigma = 1$. However, the true distribution could have a standard deviation of either $\sigma/4$ or $\sigma/2$, or σ, or $3\sigma/2$, or 2σ or 5σ. We then calculated the relative benefit of using RO over SO for different values of Γ. In both these experiments the results are similar. The relative benefit of RO is higher for higher Γ when the parameters of the assumed and the true distribution are more different. However, when the parameters are identical, the relative benefit is lower upon using a higher Γ. Thus, these results confirm that the choice of Γ should depend on the extent to which the OR manager believes the historical data to be representative of future uncertainty about case lengths. The greater the uncertainty, the greater the benefit of using a higher Γ. In the third set of experiments, we varied the functional form of the assumed and true distribution, while keeping their parameters the same. As in the previous two experiments, we found that higher Γ results in higher relative benefit when the shapes of the two distributions are more different. The results of the first set of experiments are shown in Figure 3 and of the second set of experiments are shown in Figure 4 in the online supplement.

6.4. Choice of uncertainty sets

In the RO literature, a plethora of uncertainty sets have been proposed based on different types of limit laws (see Bandi and Bertsimas (2012) for a review). We tested our approach for three different uncertainty set specifications and found that our approach delivers comparable performance ratios across these sets under different assumed distributions of Δ.

7. Conclusion

OR staffing and scheduling is a multifaceted problem. In addition to surgeons' preferences for back-to-back scheduling of their cases, OR schedules may need to account for such requirements as the pairing of specific nurse teams with specific surgeons, the availability of specialized equipment in only a subset of ORs, and the requirement to schedule certain types of cases earlier in the day (e.g. pediatric cases). The scope of OR capacity management decisions may also include pre- and post-operative processes that may cause delays in the start and/or completion of surgeries. In addition, OR scheduling practices may differ greatly from one hospital to another. It is difficult to formulate a general-purpose model that takes into account the multitude of factors and practice variations. Therefore, we chose a model that mirrors the OR capacity management practices at many community hospitals.

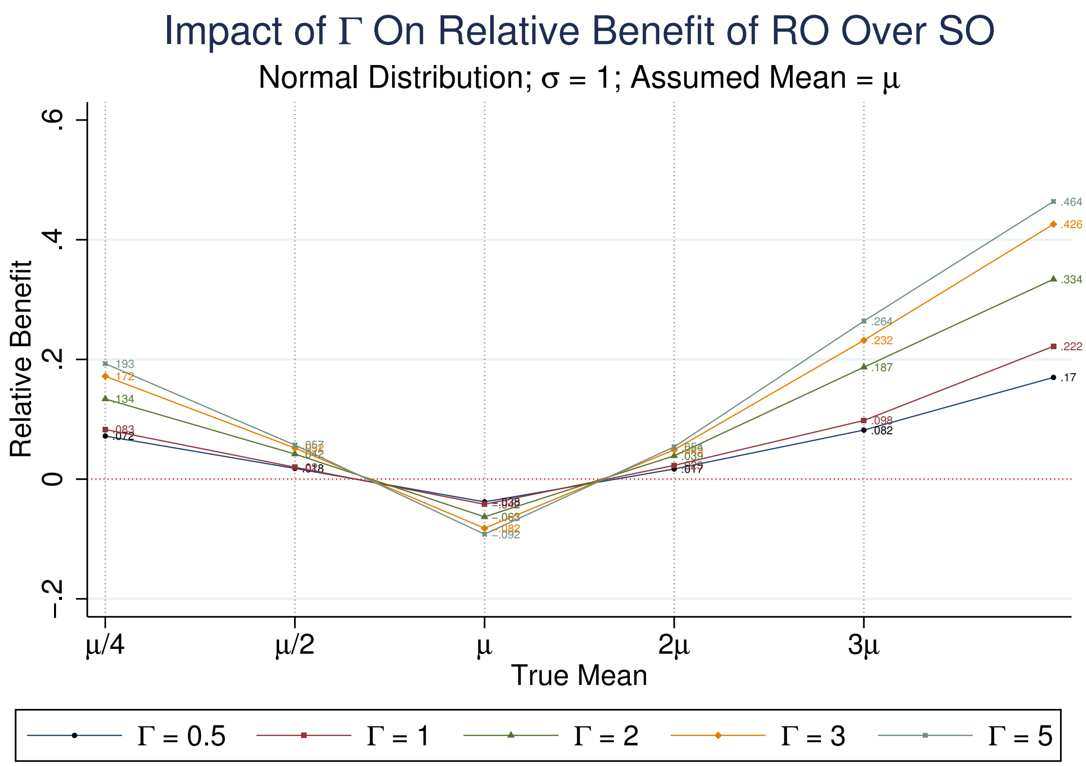

Figure 3 **The *relative benefit* of using the RO approach when true mean is different from assumed mean.**

In this sense, a limitation of our model is that it does not capture the full range of practices that might be adopted in different hospitals. Additionally, we do not model pre- and post-operative processes. We also do not consider the pairing of particular nurse teams with particular surgeons. By choosing to focus on a particular set of practices and model features, we are able to develop a tractable method for deciding baseline staffing, which must take into account the aggregate efficiency loss induced by finite shifts and discrete case lengths, and the uncertainty surrounding the actual case lengths.

In our approach, block surgeons are guaranteed the allotted amount of time, but cases are scheduled through a centralized booking station. Blocks equate to OR time, but not necessarily a particular OR for the surgeon holding the block. For some teaching hospitals that completely decentralize case scheduling, our approach may be seen as an alternative that may provide cost savings while honoring block commitments. However, because surgeons' block start times are not fixed until two or three days before the surgery day, and may depend on their realized case load, it is possible that such an approach will give rise to push back from surgeons. We recognize this implementation challenge, but also point out that recent innovations such as the bundling of payments for surgeons and hospitals provide incentives for surgeons to help lower cost and share gains from such efforts (CMS 2017).

Our approach is data driven. Specifically, we appeal to the limit laws to characterize the uncertainty set. To implement our approach the hospital needs to have access to certain amount of historical case length data that is generated by the same process that will generate future cases. Because the availability of data may vary from one hospital to another, we investigate the question of how much data will suffice in the online supplement. We find that with about 100 days of case length data, the total variation distance between the empirical and the limit distribution is less than 1%. Thus, a limitation of our approach is that a hospital that tries to implement our approach must have access to 100 or more days of representative historical data. In addition, our approach requires an expert to calculate optimal interval break points and to update them periodically. Yet another limitation is the difficulty of estimating cost parameters used in the Objective Function (8). OR managers may find it difficult to achieve consensus on the relative magnitude of surgeons' delay/idle time costs versus hospital's overtime costs.

References

Alstott, J., Bullmore, E., and Plenz, D. powerlaw: a python package for analysis of heavy-tailed distributions. *PloS one*, 9(1):e85777, 2014.

Balas, E., Ceria, S., and Cornuéjols, G. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical programming*, 58(1-3):295–324, 1993.

Balas, E., Ceria, S., and Cornuéjols, G. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, 1996.

Bandi, C. and Bertsimas, D. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical programming*, pages 1–48, 2012.

Barahona, F., Bermon, S., Günlük, O., and Hood, S. Robust capacity planning in semiconductor manufacturing. *Naval Research Logistics (NRL)*, 52(5):459–468, 2005.

Batun, S., Denton, B. T., Huschka, T. R., and Schaefer, A. J. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS journal on Computing*, 23(2):220–237, 2011.

Benchoff, B., Yano, C. A., and Newman, A. Kaiser permanente oakland medical center optimizes operating room block schedule for new hospital. *Interfaces*, 47(3):214–229, 2017.

Bertsimas, D. and Weismantel, R. *Optimization over Integers*. Dynamic Ideas, Belmont, MA, 2005.

Bertsimas, D. and Sim, M. The price of robustness. *Operations research*, 52(1):35–53, 2004.

Bertsimas, D., Goyal, V., and Sun, X. A. A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. *Mathematics of Operations Research*, 36(1):24–54, 2011.

Cha, S.-H. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(2):1, 2007.

CMS. Innovation models, 2017. URL `https://innovation.cms.gov/initiatives/index.html#views=models`. Accessed: November 28, 2017.

Coffman Jr., E. G., Csirik, J., Galambos, G., Martello, S., and Vigo, D. Bin packing approximation algorithms: survey and classification. In *Handbook of Combinatorial Optimization*, pages 455–531. Springer, 2013.

Denton, B. and Gupta, D. A sequential bounding approach for optimal appointment scheduling. *IIE Transactions*, 35(11):1003–1016, 2003.

Denton, B. T., Miller, A. J., Balasubramanian, H. J., and Huschka, T. R. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research*, 58(4-part-1):802–816, 2010.

Dexter, F., Macario, A., Traub, R., Hopwood, M., and Lubarsky, D. An operating room scheduling strategy to maximize the use of operating room block time: computer simulation of patient scheduling and survey of patients' preferences for surgical waiting time. *Anesth Analg*, 89(1):7–20, 1999a.

Dexter, F., Macario, A., and Traub, R. D. Which algorithm for scheduling add-on elective cases maximizes operating room utilization?: Use of bin packing algorithms and fuzzy constraints in operating room management. *Anesthesiology*, 91(5):1491, 1999b.

Gerchak, Y., Gupta, D., and Henig, M. Reservation planning for elective surgery under uncertain demand for emergency surgery. *Management Science*, 42(3):321–334, 1996.

Goldman, J. and Knappenberger, H. How to determine the optimum number of operating rooms. *Modern Hospital*, 111(3):114–116, 1968.

Guerriero, F. and Guido, R. Operational research in the management of the operating theatre: a survey. *Health care management science*, 14(1):89–114, 2011.

Hopp, W. J. and Lovejoy, W. S. *Hospital Operations: Principle of High Efficiency Health Care*. Pearson Education, Inc., New Jersey, 2013.

Jackson, R. L. The business of surgery. *Health management Technology*, pages 20–22, 2002.

Johnson, D. S. *Near-optimal bin packing algorithms*. PhD thesis, Massachusetts Institute of Technology, 1973.

Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R., and Graham, R. L. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4):299–325, 1974.

Kong, Q., Lee, C., Teo, C., and Zheng, Z. Scheduling arrivals to a stochastic service delivery system using copositive cones. *Operations Research*, 61(3):711–726, 2013.

Lee, C. C. and Lee, D.-T. A simple on-line bin-packing algorithm. *Journal of the ACM (JACM)*, 32(3): 562–572, 1985.

Li, F., Gupta, D., and Potthoff, S. Improving operating room schedules. *Health Care Management Science*, 19(3):261–278, 2016.

Lovejoy, W. S. and Li, Y. Hospital operating room capacity expansion. *Management Science*, 48(11): 1369–1387, 2002.

Macario, A. What does one minute of operating room cost? *Journal of Clinical Anesthesia*, 22:233–236, 2010.

May, J. H., Spangler, W. E., Strum, D. P., and Vargas, L. G. The surgical scheduling problem: Current research and future opportunities. *Production and Operations Management*, 20(3):392–405, 2011.

Mittal, S., Schulz, A. S., and Stiller, S. Robust appointment scheduling. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, page 356, 2014.

Nolan, J. P. Numerical calculation of stable densities and distribution functions. *Communications in statistics. Stochastic models*, 13(4):759–774, 1997.

Seiden, S. S. On the online bin packing problem. *Journal of the ACM (JACM)*, 49(5):640–671, 2002.

Véricourt, F. d. and Jennings, O. B. Nurse staffing in medical units: A queueing perspective. *Operations Research*, 59(6):1320–1331, 2011.

Yankovic, N. and Green, L. V. Identifying good nursing levels: a queuing approach. *Operations research*, 59 (4):942–955, 2011.

Yao, A. C.-C. New algorithms for bin packing. *Journal of the ACM (JACM)*, 27(2):207–227, 1980.

A. Proof of Theorem 2

Our proof technique is based on the proof of Theorem 3 in Lee and Lee (1985). We consider an arbitrary sequence of cases $L \in \mathcal{U}(\Gamma)$ that may need to be scheduled one-by-one on a surgery day. The sequence L consists of m cases. Let $\hat{z}_i$ be the fraction of cases whose planned lengths lie in ith interval. These intervals are defined by an interval classification algorithm ($\mathcal{A}$) whose interval breakpoints are denoted by $t_i \in (0,1],\ i=1,\cdots,K$. We consider the set of all possible constant-space online algorithms and compare them with respect to the RCR criterion. A constant-space algorithm does not keep more than a finite number of bins, say J, open at any time during its execution. This is an important tractability requirement because RCR is an asymptotic performance ratio.

Given the setup above, we calculate the maximum number of bins needed by $\mathcal{A}$ for an arbitrary L of size m. We next calculate the maximum number of bins that an **optimal** constant-space algorithm will require for an arbitrary L. Then, we take the ratio of the two quantities and its limit as $m \to \infty$. The limiting ratio is shown to be 1, completing the proof. In these arguments, we assume that $t_{K+1}=0$ for ease of exposition. In practice, there is a finite minimum case length ϵ, which may vary from one hospital to another. However, since the fraction of jobs lying in the interval $[0,\epsilon)$ for such a hospital is deterministically zero, it is easy to see that setting t_{K+1} to 0 or ϵ leads to the same scheduling algorithm and performance. Therefore, without loss of generality, we can assume that $t_{K+1}=0$.

Consider first the number of bins that an IC algorithm will need. Let us denote this quantity by $\Phi_{\mathcal{A}}(L)$. Recall that L has m total cases. In the context of the IC algorithm, a type-i case's normalized length is bounded above by t_i. Therefore, at least $\lfloor 1/t_i \rfloor$ such cases can be fitted into a unit-sized bin. There are $m\hat{z}_i$ type-i cases, implying that the maximum number of bins that an IC algorithm for an arbitrary L of size m equals

$$\Phi_{\mathcal{A}}(L) = \sum_{i=1}^{K} \lceil m \cdot \hat{z}_i / \lfloor 1/t_i \rfloor \rceil. \tag{28}$$

Note that for an interval classification algorithm, the order of arrivals of the customers does not matter. This is because cases are assigned to the interval-appropriate bins. Moreover, $\mathcal{A}$ does not have more than K bins open at any time. Therefore, so long as $K \le J$, the algorithm $\mathcal{A}$ meets the constant-space requirement.

We next consider the performance of an optimal constant-space online algorithm $\mathcal{A}^*$ and the worst case performance ratio $\chi(\mathcal{U}(\Gamma))$ given by

$$\chi(\mathcal{U}(\Gamma)) = \max_{L \in \mathcal{U}(\Gamma)} \frac{\Phi_{\mathcal{A}^*}(L)}{\Phi_{\mathcal{A}}(L)}. \tag{29}$$

By optimality of $\mathcal{A}^*$, we know that the $\chi(\mathcal{U}(\Gamma)) \le 1$. We next obtain a lower bound on the performance ratio $\chi(\mathcal{U}(\Gamma))$. We do this by considering a particular order of arrivals, where we assume that items arrive in the "decreasing" order of their sizes. This implies all the $m\hat{z}_1$ Type-1 items arrive, followed by $m\hat{z}_2$ of Type-2 items, and so on. Given that it is constant space, $\mathcal{A}^*$ can only keep up to J bins open while packing items. With this setup, the Type-1 items in the worst-case require at least $\lceil m \cdot \hat{z}_1 / \lfloor 1/t_1 \rfloor \rceil$ bins. For Type-2 items, at most $\lfloor 1/t_2 \rfloor$ items can be fitted in each bin. Therefore, an algorithm that can keep up to J bins open will need at least $\lceil (m \cdot \hat{z}_2 - J \cdot \lfloor 1/t_2 \rfloor)/\lfloor 1/t_2 \rfloor \rceil$ bins for the Type-2 items. Continuing in this fashion, the total number of bins required by $\mathcal{A}^*$ is at least equal to

$$\begin{aligned} \Phi_{\mathcal{A}^*}(L) &\geq \lceil m \cdot \hat{z}_1 / \lfloor 1/t_1 \rfloor \rceil + \textstyle\sum_{i=2}^{K} \lceil (m \cdot \hat{z}_i - J \cdot \lfloor 1/t_i \rfloor)/\lfloor 1/t_i \rfloor \rceil \\ &= \textstyle\sum_{i=1}^{K} \lceil m \cdot \hat{z}_i / \lfloor 1/t_i \rfloor \rceil - \sum_{i=2}^{K} \lfloor \frac{J \cdot \lfloor 1/t_i \rfloor}{\lfloor 1/t_i \rfloor} \rfloor \\ &= \Phi_{\mathcal{A}}(L) - (K-1)J \end{aligned} \tag{30}$$

Furthermore, we have

$$\chi(\mathcal{U}(\Gamma)) \;\geq\; \frac{\Phi_{\mathcal{A}^*}(L)}{\Phi_{\mathcal{A}}(L)} \;\geq\; 1 - \frac{(K-1)J}{\Phi_{\mathcal{A}}(L)}. \tag{31}$$

Putting together the lower and upper bounds on $\chi(\mathcal{U}(\Gamma))$, we obtain

$$1 - \frac{(K-1)J}{\Phi_{\mathcal{A}}(L)} \leq \chi(\mathcal{U}(\Gamma)) \leq 1. \tag{32}$$

Because $\Phi_{\mathcal{A}}(L) \to \infty$, as $m \to \infty$, the quantity $1 - \frac{(K-1)J}{\Phi_{\mathcal{A}}(L)} \to 1$, which implies that $\chi(\mathcal{U}(\Gamma)) \to 1$, completing the proof. □

B. Proof of Theorem 3

Let the binary decision variable x_{ij} determine if the i^{th} interval breakpoint is equal to $1/j$. That is

$$x_{ij} = 1 \quad \text{if } \tau_i = 1/j, \text{ and } 0 \text{ otherwise.} \tag{33}$$

Let $\hat{z}_i$ be the fraction of cases whose planned lengths lie in ith interval. Suppose, $\tau_i = 1/r$, and $\tau_{i+1} = 1/s$, and let f_q be the frequency of items in the Harmonic interval $(1/(q+1), 1/q]$. Then $\hat{z}_i = \sum_{q=r+1}^{s} f_q$, and $\hat{z}_i \tau_i = \frac{1}{r} \cdot \sum_{q=r+1}^{s} f_q$. That is, each $\hat{z}_i$ is the sum of item counts in contiguous Harmonic intervals, and f_q's are independent of τ_i's. The events $\{\tau_i = 1/r\}$ and $\{\tau_{i+1} = 1/s\}$ are equivalent to $\{x_{i,r} = 1, x_{i+1,s} = 1, x_{i,j} = 0 \;\forall\; j \neq r, \text{ and } x_{i+1,j} = 0, \;\forall\; j \neq s\}$, which helps us transform $\hat{z}_i$ and $\hat{z}_i t_i$ as follows.

$$\hat{z}_i = \sum_{s=r+1}^{N} \sum_{r=1}^{N} \sum_{q=r+1}^{s} f_q \cdot x_{i,r} \cdot x_{i+1,s}, \text{ and } \hat{z}_i \tau_i = \sum_{s=r+1}^{N} \sum_{r=1}^{N} \sum_{q=r+1}^{s} \frac{1}{r} \cdot f_q \cdot x_{i,r} \cdot x_{i+1,s}. \tag{34}$$

Furthermore, the variables x_{ij}'s also satisfy monotonicity and assignment constraints. The monotonicity constraint required to model $t_i \leq t_{i-1}$ implies that if the $(i-1)^{th}$ interval breakpoint t_{i-1} does not take values from the set $\{1, 1/2, 1/3, \ldots, 1/(r-1)\}$, then the i^{th} breakpoint t_i cannot take the value $1/r$. This is modeled by the constraint $x_{i,r} \leq \sum_{s=1}^{r-1} x_{i-1,s}, \; i \geq 2$.

The assignment constraints require that each t_i be assigned to one of the harmonic breakpoints ($1/j$ for some j), and that every harmonic breakpoint $1/j$ can be assigned to at most one breakpoint. These are modeled by $\sum_j x_{ij} = 1 \ \forall \ i = 1, \cdots, K,$ and $\sum_i x_{ij} \leq 1 \ \forall \ j = 1, \cdots, N.$ Finally, the optimization problem with these variables, and equivalent to Problem (8) in Sec. 3.4, is

$$\min_{\{\boldsymbol{x}\}} \max_{\{\hat{z}_i \in \mathcal{U}(\Gamma)\}} \quad \sum_{i=1}^{K} \sum_{s=r+1}^{N} \sum_{r=1}^{N} (\sum_{q=r}^{s} f_q) x_{i,r} x_{(i+1),s} \tag{35a}$$

$$\text{subject to} \quad \sum_j x_{ij} = 1 \qquad i = 1, \cdots, K \tag{35b}$$

$$\sum_i x_{ij} \leq 1 \qquad j = 1, \cdots, N \tag{35c}$$

$$x_{i,r} \leq \sum_{s=1}^{r-1} x_{i-1,s} \qquad \forall i \geq 2, \ \forall r = 1, \ldots, N \tag{35d}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j. \tag{35e}$$

Note that, in Problem (35), the objective function involves a product of binary decision variables. We next transform this problem into a mixed-integer linear program. To do this, we define new variables $y_{i,r,s}$ to model the product $x_{ir} x_{i+1,s}$ by including the following constraints $y_{i,r,s} \leq x_{i,r}, \quad y_{i,r,s} \leq x_{i+1,s}, \quad y_{i,r,s} \geq x_{i,r} + x_{i+1,s} - 1$. It can be shown that the above three inequalities form a convex hull of the non-linear constraint $y_{i,r,s} = x_{ir} x_{i+1,s}$. With this transformation, we get

$$\min_{\{\boldsymbol{x},\boldsymbol{y}\}} \max_{\{\hat{z}_i \in \mathcal{U}(\Gamma)\}} \quad \sum_{i=1}^{K} \sum_{s=r+1}^{N} \sum_{r=1}^{N} (\sum_{q=r}^{s} f_q) y_{i,r,s} \tag{36a}$$

$$\text{subject to} \quad \sum_j x_{ij} = 1 \qquad i = 1, \cdots, K \tag{36b}$$

$$\sum_i x_{ij} \leq 1 \qquad j = 1, \cdots, N \tag{36c}$$

$$x_{i,r} \leq \sum_{s=1}^{r} x_{i-1,s} \qquad \forall i \geq 2, \quad \forall r = 1, \ldots, N \tag{36d}$$

$$y_{i,r,s} \leq x_{i,r} \qquad \forall i \geq 2, \quad \forall r, s = 1, \ldots, N \tag{36e}$$

$$y_{i,r,s} \leq x_{i+1,s} \qquad \forall i \geq 2, \quad \forall r, s = 1, \ldots, N \tag{36f}$$

$$y_{i,r,s} \geq x_{i,r} + x_{i+1,s} - 1 \qquad \forall i \geq 2, \quad \forall r, s = 1, \ldots, N \tag{36g}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j. \tag{36h}$$

Finally, we dualize the inner-optimization problem to obtain

$$\min_{a,b,\boldsymbol{x},\boldsymbol{y} \in \mathcal{P}} \quad a(\Gamma M^{1/\alpha} + \sum_i \frac{\mu_i}{\sigma_i}) - b(-\Gamma M^{1/\alpha} + \sum_i \frac{\mu_i}{\sigma_i}) \tag{37a}$$

$$\text{subject to} \quad \frac{a - b}{\sigma_q} \geq c_q, \qquad q = 1, \cdots, N. \tag{37b}$$

where a and b are the dual variables and $c_q = \sum_{i=1}^{K} \sum_{s=q+1}^{N} \sum_{r=1}^{q} y_{i,r,s}$. This concludes the proof. $\square$

C. Proof of Theorem 4

Recall that L is an arbitrary finite sequence of cases with normalized case lengths $\{p_1, \cdots, p_m\}$, $\Phi_{\mathcal{A}}(L)$ denotes the number of bins required by an IC algorithm to pack the sequence L, and $OPT(L)$ denote the number of bins used by an optimal offline algorithm. Suppose $OPT(L) = n$. Without loss of generality, we may assume that each bin in the optimal packing is full. To see this, suppose the j^{th} bin in the optimal packing is not full and has free space $1 - x_j$. Then, by adding a piece of size $1 - x_j$ to the end of our sequence the cost of the optimal solution will not increase, whereas the cost to the online algorithm will not decrease. The resulting performance ratio may not therefore be the largest ratio across all L of a fixed size. Hence, to analyze the worst case bound on the performance of the algorithm, we assume all the bins in the optimal packing are full.

Let β_n denote the finite performance ratio of our IC algorithm, and let $\hat{z}_i$ be the fraction of cases whose planned lengths lie in ith interval. Then,

$$\beta_n = \max_{\{L|OPT(L)=n\}} \frac{\Phi_{\mathcal{A}}(L)}{n}. \tag{38}$$

Using (28) to substitute the value of $\Phi_{\mathcal{A}}(L)$ and noticing that the optimal breakpoints $\tau_1^*, \tau_2^*, \ldots, \tau_K^*$ are a subset of harmonic breakpoints, we obtain

$$\Phi_{\mathcal{A}}(L) = \sum_{i=1}^{K} \lceil m \cdot \hat{z}_i / \lfloor 1/\tau_i^* \rfloor \rceil = \sum_{i=1}^{K} \lceil m \cdot \hat{z}_i \tau_i^* \rceil \le \sum_{i=1}^{K} m \cdot \hat{z}_i \tau_i^* + K, \tag{39}$$

where the last inequality follows from the fact that $\lceil x \rceil \le x + 1$, for any real x.

By the assumption that all the bins in the optimal solution are full, we also have $\sum_{i=1}^{m} p_i = n$. This implies that, assuming all the items in interval $(\tau_i^*, \tau_{i+1}^*]$ take their smallest possible value τ_{i+1}^*, we must have

$$\sum_{k=1}^{K} m \cdot \hat{z}_k \tau_{k+1}^* \le \sum_{i=1}^{m} p_i = n. \tag{40}$$

Therefore, combining Equations (39) and (40), we may compute β_n by solving:

$$\beta_n \quad = \quad \max_{(\hat{z}_1,\ldots,\hat{z}_K)\in\mathcal{U}(\Gamma)} \frac{1}{n} \cdot \left\{ \sum_{i=1}^{K} m \cdot \hat{z}_i \tau_i^* + K \right\} \quad \text{s.t.} \quad \sum_{k=1}^{K} m \cdot \hat{z}_k \tau_{k+1}^* \le n.$$

Upon letting $n, m \to \infty$, and dividing both objective and constraint by n, we obtain the equivalent form as:

$$\lim_{n\to\infty} \beta_n \quad = \quad \max \sum_{i=1}^{K} \hat{z}_i \tau_i^* \quad \text{s.t.} \quad \sum_{k=1}^{K} \hat{z}_k \tau_{k+1}^* \le 1, \ (\hat{z}_1, \ldots, \hat{z}_K) \in \mathcal{U}(\Gamma).$$

This concludes the proof. □

D. Computational Tractability of Phase 1 and Phase 2

In what follows, we will use e_i to denote the vector with a 1 in the i^{th} component and zeros everywhere else.

Structure of Phase-1 Optimization Problem (5)

In order to demonstrate the structure we will use $\boldsymbol{\xi}$ to denote the vector of all the variables $\boldsymbol{x}, \boldsymbol{y}$. In particular, $\boldsymbol{\xi} = \left\{ \{x_{i,j}\}_{i=1,\ldots,K; j=1,\ldots,N}, \{y_{i,r,s}\}_{i=1,\ldots,K; r,s=1,\ldots,N} \right\}$. Note that the size of this vector is $2NK + KN^2$.

- *Structure of constraints (6b,6c)*: These constraints correspond to an assignment problem and are known to be tight.
- *Structure of constraints (6d)*: We show that these constraints have the consecutive 1's property. In particular, we will show that each row corresponding to these constraints in matrix A will only have +1's and -1's occuring consecutively. Recall that the size of each row is $2NK + KN^2$. For a fixed value of r, the constraint is given by $x_{i,r} \leq \sum_{s=1}^{r} x_{i-1,s}$. This corresponds to a +1 coefficient for $x_{i,r}$ and consecutive -1s for $x_{i-1,s}$ for $s = 1, \ldots, r$. Therefore, the corresponding row in matrix A will consist of +1 in position $(i-1)*K + r$ and consecutive -1s in positions $(i-2)*K + 1$ to $(i-2)*K + r$.
- *Structure of constraints (6e)*: We show that these constraints have the consecutive 1's property. For a fixed value of w, the constraint is given by $\sum_{i=1}^{K} \sum_{s=w+1}^{N} \sum_{r=1}^{w} y_{i,r,s}$. This corresponds to a +1 coefficient for $y_{i,r,s}$s present in the summation. Therefore, for a fixed value of w, the corresponding row in matrix A will consist of consecutive +1s in the following positions: for each pair (i,r) with $i = 1, \ldots, K; r = 1, \ldots, w$ from position $(i-1)N^2 + (r-1)N + w + 1$ to $(i-1)N^2 + (r-1)N + N$.
- *Structure of constraints (6f)–(6h)*: These set of constraints are a linear representation of the constraints: $y_{i,r,s} = x_{i,r} x_{i+1,s}$. In order to show the tightness of constraints (6f)–(6h), we use the following general result:

$$\text{Convex hull}\left\{(x,y,z)\,|z = xy\right\} = \left\{(x,y,z)\,|z \leq x,\ z \leq y,\ z \geq x + y - 1,\ z \geq 0.\right\} (= P_{\text{AND}}). \tag{41}$$

This shows that the set of constraints denoted above by P_{AND} are a convex hull of set of points satisfying $z = xy$.

Structure of Phase-2 Optimization Problem (8)

In optimization problem 8, a subset of constraints have the consecutive 1's structure which gives rise to practically good performance. In particular, the *Surgery sequencing constraints* (Eqs. (9)–(13)) and the *Idle time control constraints* (Eqs. (20)–(**??**)) have the consecutive 1's property, while the remanining constraints do not. We exploit this structure and solve this problem using a cutting plane approach. In particular, we use a standard algorithm used in Barahona et al. (2005) which is based on a cutting plane algorithm developed in Balas et al. (1996, 1993). In what follows, we demonstrate the consecutive 1s property in each constraint in the same manner as in Problem 5.

We proceed as in the case of Problem 5, and will use $\boldsymbol{\omega}$ to denote the vector of all the variables $\boldsymbol{\xi}, \boldsymbol{o}, \boldsymbol{\chi}$. In particular, $\boldsymbol{\omega} = \{\{\xi_{h,j,k,i}\}, \{o_{h,j,k,i}\}, \{\chi_{h,j,k,i}\} \quad \forall h,i,j,k\}$. Note that the size of this vector is $3\left(\sum_{h=1,\ldots,M} N_h\right)^2$. We further show that these constraints have the consecutive 1s property:

(1) *Structure of constraints (9)*: The constraint involves a summation over all the χ variables. This corresponds to a $+1$ coefficient for $\chi_{h,j,k,i}$ and therefore the row in matrix A will have a $+1$ consecutively in the following positions: $2\left(\sum_{h=1,\ldots,M} N_h\right)^2 + 1$ to $3\left(\sum_{h=1,\ldots,M} N_h\right)^2$.
(2) *Structure of constraints (10)–(13)*: In constraint (10), there are $\sum_{k=1}^{M} N_k$ such constraints for different values of (k,i) $\forall k = 1,\ldots,M \quad \forall i = 1,\ldots,N_k$. For a fixed value of (k,i), the constraint involves a summation over all the $\chi_{h,j,k,i}$ variables for different values of (h,j). This corresponds to a $+1$ coefficient for $\chi_{h,j,k,i}$ and therefore the row in matrix A will have a $+1$ consecutively in the following positions: $(\sum_{s=1}^{h-1} N_s + j - 1)\left(\sum_{h=1,\ldots,M} N_h\right) + 1$ to $(\sum_{s=1}^{h-1} N_s + j)\left(\sum_{h=1,\ldots,M} N_h\right)$. Constraints (11)–(13) have a similar structure.
Finally, the Idle time control constraints in (Eqs. (20)–(**??**)) have the same structure as the Surgery sequencing constraints.

In order to further demonstrate this tractability, we present results from computational experiments performed on multiple instances of optimization problems (5) and (8). The computations were performed using the Concert Technology of CPLEX 12.4, a state-of-the-art professional MIP solver on a Ubuntu based desktop computer (Intel Core 2 Duo CPU, 3.0GHz, 8GB of RAM). We generated 100 random instances of optimization problems, each of sizes $N = 10, 50, 100, 200, 500$. The computing time grows from 0.5 seconds for $N = 10$ to 135 seconds for $N = 500$, and are therefore practical. These experiments were performed only to compare the computing times and the values of N chosen do not necessarily have any practical relevance.

E. More Robustness Tests

In this section, we present additional computational results that demonstrate how using an RO approach provides better performance when there may be errors in estimating distributional parameters. The experimental setup is the same as in Section 6.2, and we calculate the *relative benefit* as a measure of performance. It is defined as follows:

$$\text{Relative Benefit } = \frac{\text{Cost of Stochastic Optimal Schedule} - \text{Final cost of our Algorithm}}{\text{Cost of Stochastic Optimal Schedule}}. \tag{42}$$

We consider two scenarios. In both cases, the functional form of the assumed distribution of Δ is correct. It is furthermore assumed $\mathbb{F}_\Delta$ has a normal distribution with mean μ and standard deviation σ. The first scenario incorrectly estimates σ, and the second scenario incorrectly estimates μ.

F. Robustness to the Choice of Uncertainty Sets

In this section, we consider other choices of uncertainty sets being considered in the RO literature. In particular, we tested our approach for the following uncertainty sets.

1. *Stationary* uncertainty set, $\mathcal{U}^1$ given by $\mathcal{U}^1 = \left\{ (\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m) \,\middle|\, -\Gamma m^{1/\alpha} \le \frac{\sum_{i=1}^m \tilde{p}_i - m\mu}{\sigma} \le \Gamma m^{1/\alpha} \right\}$.
2. *Heavy-tailed Mean absolute deviation* uncertainty set, $\mathcal{U}^2$ given by $\mathcal{U}^2 = \left\{ (\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_m) \,\middle|\, -\Gamma m^{1/\alpha} \le \sum_{i=1}^m \frac{\tilde{p}_i - p_i}{p_i} \le \Gamma m^{1/\alpha} \right\}$.

Assumed Parameters $\mu=1$	True Parameters; $\mu=1$		
	$\sigma=0.5$	$\sigma=1$	$\sigma=2$
$\sigma/4$	0.108	0.134	0.196
$\sigma/2$	0.0357	0.042	0.068
$3\sigma/2$	0.0282	0.039	0.062
2σ	0.141	0.187	0.261
5σ	0.247	0.334	0.542

Assumed Parameters $\sigma=$ true value	True Parameters; $\mu=1$		
	$\sigma=0.5$	$\sigma=1$	$\sigma=2$
$\mu/4$	0.178	0.22	0.335
$\mu/4$	0.053	0.064	0.09
2μ	0.176	0.242	0.392
3μ	0.312	0.416	0.58
5μ	0.623	0.586	0.712

Table 9 ***Our Algorithm* vs *Cost of Stochastic Optimal Schedule*: The *relative benefit* under the same mean but different standard deviations (left); and same standard deviation but different means (right).**

3. *Mean absolute deviation* uncertainty set, $\mathcal{U}^3$ given by $\mathcal{U}^3 = \left\{(\tilde{p}_1,\tilde{p}_2,\ldots,\tilde{p}_m)\,\middle|\, -\Gamma\sqrt{m} \le \sum_{i=1}^{m}\frac{\tilde{p}_i-p_i}{p_i} \le \Gamma\sqrt{m}\right\}$.

The first is based on the set proposed in Bertsimas et al. (2011) and is known to have favourable geometric properties. The second is also proposed in Bertsimas et al. (2011) and captures the heavy tailed nature using the heavy tail coefficient α. The third set is proposed in Bandi and Bertsimas (2012). As before, we compare the performance of our approach with an offline optimal solution. The results are reported in Table 10, where we pick $\Gamma = 2$. By examining each row in this table, we observe that the performances are very similar across different choices of the uncertainty sets for each assumed distribution of Δ.

Distribution of Δ ($\mathbb{F}_\Delta$)	$\mathcal{U}^1$	$\mathcal{U}^2$	$\mathcal{U}^3$
Normal($\mu=10,\sigma=2$)	1.29,0.3,1.58	1.3,0.31,1.63	1.27,0.29,1.61
Exponential($\lambda=1$)	1.29,0.29,1.61	1.34,0.32,1.63	1.29,0.29,1.59
Standard LogNormal	1.39,0.34,1.58	1.41,0.28,1.61	1.43,0.26,1.59
Standard Pareto ($\alpha=1.7$)	1.29,0.32,1.68	1.4,0.32,1.66	1.38,0.32,1.72
Real Data ($\mathbb{F}_\Delta^e$)	1.21,0.21,1.38	1.2,0.22,1.46	1.18,0.22,1.42

Table 10 **Relative performance of different uncertainty sets. Reported statistics: the mean, the standard deviation, and the 95^{th} percentile of the performance ratio.**

G. Convergence to the Limit Distribution

Our approach of constructing uncertainty sets is based on appealing to various limit laws. For light tailed distributions, we use the central limit theorem, and for heavy tailed distributions, we use the Stable Limit law. While for light tailed distributions, the limit distribution is reached swiftly (error rates of less than 1% are observed with sample size of ≈ 30), heavy tailed distributions require more samples. In what follows, we present a comparison of how quickly the stable limit distribution is reached for various parameters, by calculating the *total variation* (TV) distance between the empirical distribution and the limit distribution. TV distance is a common metric to represent how close different distributions are (see Cha (2007)). We plot the TV distance for different distributions as a function of the sample size in Figure 10. We observe that even for a heavy tail coefficient of 1.5, the

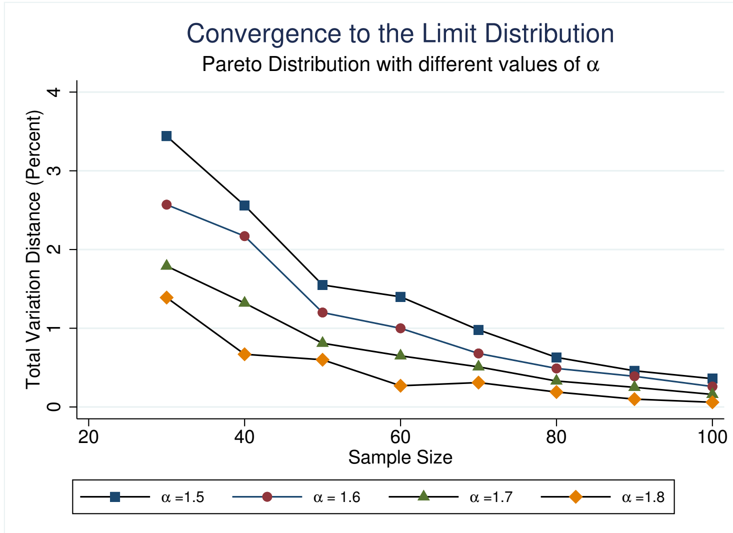

Figure 4 **Convergence to the Limit Distribution for different heavy tail coefficients α.**

stable limit distribution is reached within an error of 0.5% with 100 samples. This supports our claim that hospitals do not need a lot of historical data to obtain good results upon using our approach.

H. Choice of Γ

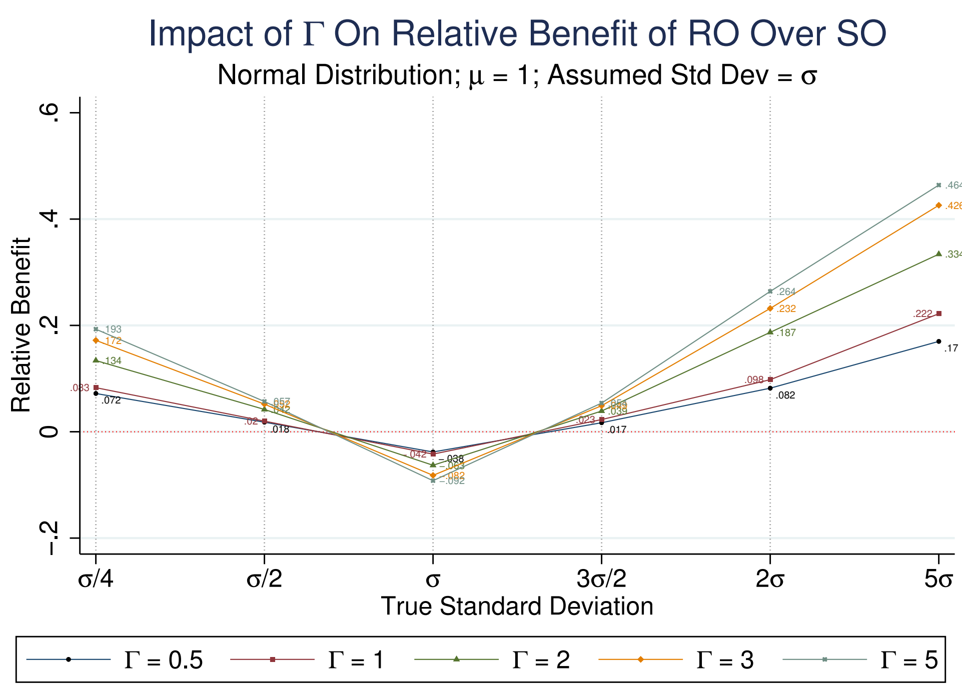

Figure 5 **The *relative benefit* of using the RO approach when true standard deviation is different from assumed standard deviation.**