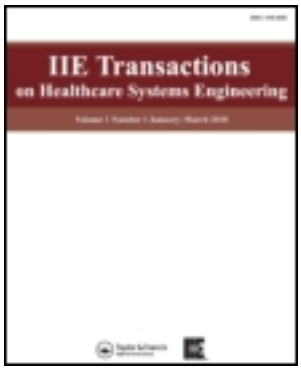

IIE Transactions on Healthcare Systems Engineering
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uhse20

A stochastic dynamic program for the single-day surgery scheduling problem

William L. Herring [a] & Jeffrey W. Herrmann [b]

[a] Univerisity of Maryland, Applied Mathematics & Statistics and Scientific Computation, Mathematics Building, College Park, MD, 20742, USA

[b] University of Maryland, Department of Mechanical Engineering and Institute for Systems Research, College Park, MD, USA

Available online: 16 Dec 2011

To cite this article: William L. Herring & Jeffrey W. Herrmann (2011): A stochastic dynamic program for the single-day surgery scheduling problem, IIE Transactions on Healthcare Systems Engineering, 1:4, 213-225

To link to this article: http://dx.doi.org/10.1080/19488300.2011.628638

IIE Transactions on Healthcare Systems Engineering (2011) **1**, 213–225

ISSN: 1948-8300 print / 1948-8319 online
DOI: 10.1080/19488300.2011.628638

A stochastic dynamic program for the single-day surgery scheduling problem

WILLIAM L. HERRING[1,*] and JEFFREY W. HERRMANN[2]

[1]*Univerisity of Maryland, Applied Mathematics & Statistics and Scientific Computation, Mathematics Building, College Park, MD 20742, USA*
E-mail: wherring@math.umd.edu
[2]*University of Maryland, Department of Mechanical Engineering and Institute for Systems Research, College Park, MD, USA*

Received November 2010 and accepted September 2011.

Scheduling elective surgeries involves sequential decision-making on the part of the operating room (OR) manager, who must continually balance the costs of deferring waiting cases and blocking higher-priority cases. While surgery scheduling has received extensive treatment in the literature, this paper presents the first modeling approach to capture this aspect of the process while incorporating block schedules, block release policies, and surgical waiting lists. The result is a stochastic dynamic programming formulation for the evolution of the schedule for a single day in an OR suite over the days leading up to the day of surgery. A general formulation is presented and theoretical results are obtained for a single-room version. These results demonstrate that optimal waiting list decisions for a single OR follow a threshold policy that preserves a desired amount of OR time for the remaining demand from the room's allocated surgical specialty. An algorithm for determining the optimal thresholds is presented, followed by computational results.

Keywords: healthcare management, operating room, dynamic programming, optimization, scheduling

1. Introduction

The health care system has been the focus of increasing attention from operations researchers and management scientists in recent years. The growth in this research area is motivated by increasing costs and the rising demand for health care services and is facilitated by the improved quality and availability of data generated by the system. Among the problems studied in the healthcare system, the problem of scheduling surgical patients into hospital operating room (OR) suites benefits from a particularly robust literature. This interest in surgery scheduling is intensified by the key role that OR scheduling plays in determining hospital occupancy levels and because the OR is the most resource-intensive and profitable unit of a hospital (Macario *et al.*, 1995; McManus *et al.*, 2003). At its core, the surgery scheduling problem, in all its variations, involves the allocation of a fixed amount of resources (ORs, hospital staff) under uncertain demand (see Cardoen *et al.*, 2010, for a thorough review). Like other scheduling problems, surgery scheduling approaches hope to make more efficient use of existing resources. However, as we will see below, the large number of stakeholders and the contentious nature of surgery scheduling introduce complexities that help to distinguish it from other scheduling problems.

1.1. *Literature review*

The majority of hospitals schedule their OR suites using cyclic master, or block, surgery schedules, in which available OR space is assigned to specific surgical specialties. For hospitals using block schedules, the literature on surgery scheduling describes the problem as consisting of three stages: (1) determining the amount of OR time to allocate to various surgical specialties, (2) creating a block schedule implementing the desired allocations, and (3) scheduling individual patients into available time (Blake and Donald, 2002; Santibañez *et al.*, 2007; Testi *et al.*, 2007). The first stage is referred to as case mix planning, and decisions at this stage typically reflect the long-term strategic goals of hospital management, such as meeting the demand for surgical specialties' services, achieving desired levels of patient throughput, or maximizing revenue (Blake and Carter, 2002; Gupta, 2007; Santibañez *et al.*, 2007; Testi *et al.*, 2007). The second and third stages represent medium- and short-term operational decisions, respectively, but differ markedly in their objectives. Block scheduling models have traditionally focused on implementing desired allocation levels, but are moving toward a focus on leveling hospital

*Corresponding author

bed occupancy and minimizing overcapacity (Beliën and Demeulemeester, 2007; Blake and Donald, 2002; Price *et al.*, 2011; van Oostrum *et al.*, 2008). Research on individual patient scheduling, including patient selection, room placement, and sequencing, aims to minimize patient delays and maximize OR utilization (Denton *et al.*, 2007; Guinet and Chaabane, 2003). The large number of stakeholders involved in surgery scheduling has also motivated a number of multi-objective models for both block scheduling and individual patient scheduling (Beliën *et al.*, 2009; Blake and Carter, 2002; Ozkarahan, 2000).

A case study of the surgery scheduling system at a large, university hospital revealed that a fundamental, but understudied, element of the day-to-day job of surgery scheduling is the transition from the generality of the block schedule (in which OR time is allocated to specialties) to the specificity of a completed schedule for a particular day (in which OR time is allocated to specific cases) (Herring, 2011). While block schedules are often incorporated as constraints in individual patient scheduling models (Hans *et al.*, 2008; Pham and Klinkert, 2008; Testi *et al.*, 2007), these models schedule large batches of patients all at once, rather than sequentially. As will be discussed in greater detail below, in practice, individual patients are scheduled into ORs over time as the demand for surgery is generated, resulting in a dynamic, sequential decision-making process. The studies that do consider the dynamic evolution of a surgical schedule focus on the scheduling of waiting list cases, but do so either for a limited number of cases or a limited number of days (Dexter *et al.*, 1999; Dexter and Traub, 2002; Dexter *et al.*, 2003; Dexter and Macario, 2004; Gerchak *et al.*, 1996). As a result, these studies only provide a limited picture of the dynamics of the surgery scheduling process. This paper proposes a new model for the single-day surgery scheduling problem that seeks to address these shortcomings and sheds new light on the management policies that control the transition between block and individual patient scheduling.

The stochastic dynamic programming (SDP) model that we propose has much in common with existing research on capacity allocation (Schütz and Kolisch, 2010a, b) and, in particular, airline revenue management (Brumelle and Walczak, 2003; Lee and Hersh, 1993). While the specifics of our model will be presented later, we conclude our literature review with a brief mention of what distinguishes our work from this related research. In these problems, a finite resource (OR time, seats on a flight) with a fixed expiration date (the day of surgery, the departure time for the flight) must be allocated to competing demand classes. The demand from each of the classes arrives over time, and decision-makers must decide how much of the resource to allocate to lower priority classes and how much to reserve for higher priority classes. In the existing models, arriving demand must be accepted or rejected at the moment of its arrival and rejected demand is lost. In the surgery scheduling problem, however, lower priority demand is placed on a waiting list and can be accepted at a number of different decision epochs leading up to the day of surgery. While our proposed model displays solution behavior similar to the revenue management models (particularly to Lee and Hersh, 1993), the introduction of the waiting list concept increases the complexity of the state space and thus complicates the analysis.

1.2. *Dynamics of surgery scheduling*

To place the importance of the transition between block and individual patient scheduling into the proper context, it is necessary to understand the dynamic way in which the schedule for a single day evolves. Our description of these dynamics and the resulting problem statement are based on a case study of the surgery scheduling system at the University of Maryland Medical Center (UMMC) performed in collaboration with UMMC's peri-operative services department (Herring, 2011). The literature suggests that similar systems are in place at other hospitals that use block scheduling (Dexter *et al.*, 2003; Dexter and Macario, 2004; Ozkarahan, 2000).

Block schedules are concerned primarily with elective surgery, and the demand for elective surgery arrives over the course of multiple days before the day of surgery, rather than all at once. For a given day on which an OR suite is open for surgery, the available OR time is allocated to specific surgical specialties according to a block schedule. Blocks of OR time are initially controlled exclusively by these "primary" specialties, and the primary specialties are free to choose and sequence their cases ("primary" cases) within their allocated blocks as they see fit. Specialties or surgeons that do not have allocated OR time on that day, but still wish to perform a surgery, must submit their cases ("secondary" cases) to the surgical request queue (RQ) for that day. If a primary specialty's allocated OR time has been filled, it may also submit its excess cases to the RQ. In the period leading up to the day of surgery, cases accumulate on the RQ and OR managers look to schedule these RQ cases into OR time that has not been filled by the primary specialties. In practice, there is a set day before surgery, referred to as the *block release date*, after which OR managers can begin assigning RQ cases to unfilled blocks. Before the block release date, RQ cases may not be assigned to open times. After the block release date, the *request queue decisions* determine how and when RQ cases will be assigned to open times. Together, the block release dates and RQ decisions are the primary means by which hospital administrators and the OR manager control the transition between the block and individual patient scheduling stages of surgery scheduling.

In our setting, both the demand for elective surgery and the RQ are day-specific. That is, surgeons submit cases for a specific date in the future on which they would like to perform surgery. Similarly, elective cases on a given day's RQ that are never scheduled are not automatically rolled

over to future days' RQs (although surgeons can resubmit their cases for future days). As a result, we can separate the general problem of scheduling the ORs on a rolling basis into sub-problems that consider the development of only one day's schedule.

RQ decisions are made every day, once a day, before any newly generated surgical demand has been communicated to the hospital. In practice, OR managers must manage the schedule for many future surgery days, and it is impractical for them to spend time reconsidering RQ cases too frequently. The demand for surgery is typically communicated to the hospital by the specialties once a day, usually after the surgeons have seen their patients in a clinic or made their rounds. It is also important to note that this problem does not consider the process of scheduling urgent (emergent) cases that are generated on the day of surgery. In practice, entire blocks may be reserved for urgent cases, or the block schedule may include some slack for these cases. These and other similar approaches affect the stated capacity of the ORs, which our approach will treat as given.

The dynamic aspect of scheduling individual patients emphasizes the importance of considering the joint impact of block schedules, block release dates, and RQ decision-making policies when studying single-day surgery scheduling. However, no existing work in the surgery scheduling literature addresses all three of these components simultaneously. As mentioned above, some recent papers incorporate block schedules as constraints in models that schedule a batch of individual patients all at once rather than sequentially (Hans *et al.*, 2008; Pham and Klinkert, 2008; Testi *et al.*, 2007). A pair of papers analyze different block release dates and conclude that the timing of the block release has little impact on OR efficiency (Dexter *et al.*, 2003; Dexter and Macario, 2004). However, their work, which adds single RQ cases to existing schedules at different points in the evolution of the schedules, fails to capture the potential effect of the block release on scheduling decisions made after the RQ case has been added. Other related papers consider methods for placing RQ, or add-on, cases into existing schedules, but the decisions are limited to the day before and the day of surgery, thus neglecting the role of the block release date (Dexter *et al.*, 1999; Dexter and Traub, 2002; Gerchak *et al.*, 1996). In addition to these shortcomings in the literature, our collaborators at UMMC reported a history of contention at their facility surrounding the choice of block release dates and the more general problem of how blocks of OR time are released to RQ cases.

The remainder of this paper develops and analyzes a model for the dynamic single-day surgery scheduling problem that considers the realistic elements referenced above. The immediate goal is to capture the most important relationships and develop insights into the scheduling process, particularly focusing on the often contentious nature of block release and RQ decisions. Section II offers a formal statement of the problem and a stochastic dynamic programming (SDP) formulation. Section III proposes a set of analytical results about the structure of solutions to the single OR version of the SDP that offer an intuitive new perspective on how block release and RQ policies can be made more equitable and transparent. The proof of these results suggests an algorithm for determining optimal decisions without fully evaluating the SDP. Section IV discusses this new perspective and presents computational results on the sensitivity of the optimal decisions to the input data. Finally, Section V offers some concluding remarks and indicates directions for future research. Our approach reflects numerous discussions with collaborators at UMMC, whose response to our results and their policy implications has been positive. Their feedback informs the wide range of ongoing and future collaborative research that is needed before implementation of our results at UMMC or any other facility.

2. Problem statement and formulation

2.1. *Deferral and blocking costs*

Before formally stating the dynamic single-day surgery scheduling problem (SDSSP), it is important to understand the costs associated with the evolution of a schedule for a suite of ORs. Two types of costs are associated with the process of scheduling an OR suite that uses block scheduling: (1) utilization costs related to under- and overutilization of ORs and (2) customer (surgeon and patient) satisfaction costs incurred when cases are forced to wait on the RQ or when primary cases are blocked from their allocated ORs due to RQ placements. The utilization costs are common in the literature, and others have acknowledged the existence of the satisfaction costs (Dexter and Macario, 2004), but our formulation is the first to formalize the satisfaction costs and model them as a fundamental component of the scheduling process.

Consider a scenario in which several days before the day of surgery, an OR's schedule has available time for one additional case and the RQ has a case that fits in the available time. The OR manager has the following choice: either schedule the RQ case into the available time or defer scheduling the case (and consider it again the next day, if the time is still available). Deferring the case leaves open the possibility that the OR's primary specialty will generate a case and fill the remaining time, but is undesirable because the surgeon and the patient associated with the RQ case must wait at least one more day for a decision. We define this customer satisfaction cost as a *deferral cost*. If the RQ case is scheduled (thus filling the OR) and another primary case arrives, then the OR manager has blocked the primary specialty's access to its allocated room. We define this satisfaction cost as a *blocking cost*. The balance between deferral costs and potential blocking costs informs the OR manager's decision to schedule or defer RQ cases. The stochastic decision tree in Figure 1 illustrates the

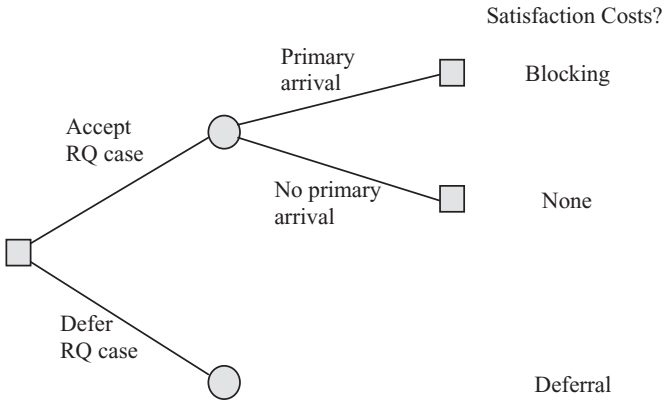

Fig. 1. Decision tree for a simple request queue scenario.

potential outcomes for this simple scenario. While these deferral and blocking costs are primarily satisfaction costs, it is important to note that their relative values may reflect a range of practical concerns, including differences in patient acuity levels, specialized resource needs, and contributions to revenue between surgical specialties.

2.2. *Problem statement*

We consider the schedule for a single day in a suite of ORs allocated to surgical specialties according to a block schedule and operating under the general policies described above for hospitals using block schedules. On each day leading up to the day of surgery, the OR manager must choose the number of RQ cases to add to each OR's schedule for the day of surgery. The manager's objective is to minimize the expected total cost of deferral and blocking penalties incurred on the days before surgery and OR underutilization on the day of surgery. Block release dates restrict the days on which RQ cases can be added to the schedule and may differ from room to room. After the block release dates, RQ decision-making policies dictate if and when RQ cases will be added to an OR's schedule.

The dynamic relationship between the arrival of surgical demand, RQ decisions, and the associated costs suggests a SDP formulation. Because block release dates serve as constraints, the optimal policy combination will use no block release dates and use the optimal decisions from the unconstrained SDP as the RQ decision-making policy. For this reason, the formulation and analysis that follow consider an SDP unconstrained by block release dates. Because block release dates are used in practice (to protect OR time allocated to primary specialties and to reduce the OR manager's workload by limiting the number of future days that must be considered), the relationship between our unconstrained optimal solutions and traditional block release dates will be discussed. This comparison leads to a fundamentally new perspective on optimal block release and RQ decision-making.

A few final comments are needed before presenting our initial SDP formulation of the dynamic SDSSP. While a general formulation incorporating multiple ORs and varying case durations can be given (Herring, 2011), the general formulation is computationally burdensome and yields little insight into the nature of the optimal decisions. Therefore, the formulation presented in this paper represents the problem for a scenario where all cases have the same (unit) duration. The subsequent analysis of the optimal solution structure further focuses on the problem with a single OR (SDSSP1). Our decision variables reflect the number of cases to add to an OR's schedule on each day leading up to the day of surgery. The decision not to model a binary decision variable for each case on the RQ not only helps maintain the computational tractability of the model, but also preserves a desirable measure of freedom for the OR manager in choosing exactly which RQ cases to schedule. Our analysis of SDSSP1 yields insights into the more general SDSSP and represents the first step toward understanding and optimizing more realistic versions involving multiple case durations, multiple ORs, and other complexities that an OR manager must consider in practice (see Herring and Herrmann, 2011, for an analysis of the model incorporating different case durations).

2.3. *SDP formulation*

Define the problem input data as follows:

S = number of rooms in the OR suite
N = number of days before the day of surgery on which surgical demand is generated
C = capacity of a single OR

The stochastic demand for surgery is given by the following random variables. Arrivals are associated with their specialties, and a one-to-one correspondence between rooms and specialties is assumed. From this point forward, all references to days represent the number of days before the day of surgery, with day 0 being the day of surgery. For $s = 1, \ldots, S$ and $j = 0, \ldots, N$:

T_{sj} = number of primary demand arrivals to OR s on day j
R_j = number of secondary demand arrival son day j

The utilization and customer satisfaction costs are similarly defined by day and service line. For $s = 1, \ldots, S$ and $j = 1, \ldots, N$:

h_0 = penalty for unscheduled cases left on RQ on the day of surgery
r_{s0} = penalty for unused space in OR s on the day of surgery
h_j = deferral cost for day j
r_{sj} = blocking cost for specialty s on day j

In addition to the number of days remaining until surgery, there are three types of state variables needed for the dynamic program. For $s = 1, \ldots, S$ and $j = 0, \ldots, N$:

W_j = number of cases on the RQ on day j
B_{sj} = number of blocking eligible cases in OR s on day j
C_{sj} = available space in OR s on day j

The states representing the number of cases on the RQ and the available space in each of the ORs are self explanatory. The number of blocking eligible cases in an OR reflects the presence of RQ cases added to the room on previous days that have yet to incur a blocking penalty. This auxiliary state accounts for the fact that a primary specialty's case might be blocked by a RQ placed in the room several days earlier.

Finally, the decision variables are defined by room and day. For $s = 1, \ldots, S$ and $j = 0, \ldots, N$:

x_{sj} = number of RQ cases to add to OR s on day j

With the exception of the day of surgery, the costs incurred on day j are separated into deferral costs and blocking costs. Deferral penalties are only assessed up to the number of RQ placements that are feasible. In other words, if four cases are on the RQ but there are only two open spaces, then at most two deferrals are allowed because at most two cases can be taken off the RQ. For $j = 1, \ldots, N$:

$$\begin{aligned} N_j^d &= \text{number of deferrals on day } j \\ &= \min\left(W_j, \sum_{s=1}^{S} C_{sj}\right) - \sum_{s=1}^{S} x_{sj} \end{aligned}$$

At the time the RQ decisions are made, day j's arrivals have not yet been realized, therefore the number of blocking penalties incurred in room s on day j is a random variable depending on the primary specialty's arrivals, T_{sj}. Figure 2 illustrates how the number of blocking penalties relates to these arrivals, leading to an expression for the number blocked in room s on day j. For $s = 1, \ldots, S$ and $j = 1, \ldots, N$:

$$\begin{aligned} N_{sj}^b &= \text{number blocked in OR } s \text{ on day } j \\ &= \begin{cases} 0 & \text{if } T_{sj} \le C_{sj} - x_{sj} \\ B_{sj} + x_{sj} & \text{if } T_{sj} \ge C_{sj} + B_{sj} \\ T_{sj} - \left(C_{sj} - x_{sj}\right) & \text{otherwise} \end{cases} \\ &= \max\left(0, \min\left(B_{sj} + x_{sj}, T_{sj} - C_{sj} + x_{sj}\right)\right) \end{aligned}$$

This is now sufficient information to describe the transitions between states from one day to the next. In the transition equations below, it is important to remember that primary case arrivals that exceed the available space in their allocated room are redirected to the RQ. The RQ transition equation reflects previous day's RQ status (W_j), the number of cases added to the schedule (x_{sj}), the secondary arrivals (R_j), and the number of primary arrivals that are redirected to the RQ ($\max(0, T_{sj} - C_{sj} + x_{sj})$). The blocking eligible transition equation for an OR reflects the previous day's blocking eligible cases (B_{sj}), the number of new cases added to the room (x_{sj}), and the number of blocking penalties incurred (N_{sj}^b). Finally, the transition equation for the remaining space in the OR reflects the previous day's available space (C_{sj}), the space taken up by newly scheduled RQ cases (x_{sj}), and the number of primary arrivals that will still fit in the OR ($\min(T_{sj}, C_{sj} - x_{sj})$). For $s = 1, \ldots, S$ and $j = 1, \ldots, N$:

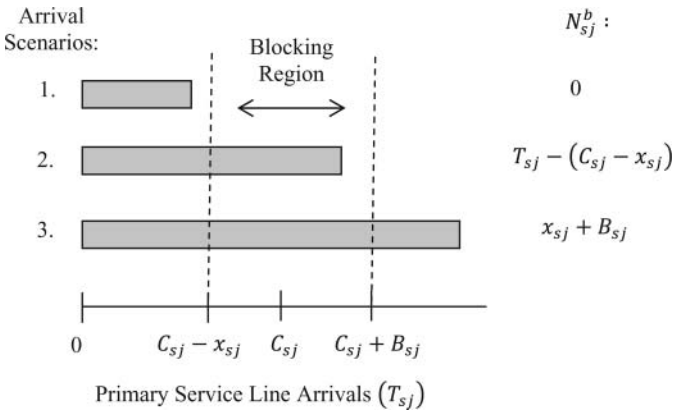

Fig. 2. Illustration of different blocking penalty scenarios.

$$\begin{aligned} W_{j-1} &= W_j - \sum_{s=1}^{S} x_{sj} + R_j + \sum_{s=1}^{S} \max\left(\mathbf{0}, T_{sj} - C_{sj} + x_{sj}\right) \\ B_{s,j-1} &= B_{sj} + x_{sj} - N_{sj}^b \\ C_{s,j-1} &= C_{sj} - x_{sj} - \min\left(T_{sj}, C_{sj} - x_{sj}\right) \end{aligned}$$

Expressing the states and decision variables across all operating rooms as vectors $\boldsymbol{B_j}$, $\boldsymbol{C_j}$, and $\boldsymbol{x_j}$, the value function for the stochastic dynamic program for days $j = N, \ldots, 1$ can be defined as:

$$\begin{aligned} V_j\left(W_j, \boldsymbol{B_j}, \boldsymbol{C_j}\right) &= \text{minimum expected remaining cost from} \\ &\quad \text{state } \left(\mathrm{W}_j, \boldsymbol{B_j}, \boldsymbol{C_j}\right) \text{ on day } j \\ &= \min_{x_j} \left\{ h_j N_j^d + \sum_{s=1}^{S} r_{sj} E\left[N_{sj}^b\right] \right. \\ &\quad \left. + E\left[V_{j-1}\left(W_{j-1}, \boldsymbol{B_{j-1}}, \boldsymbol{C_{j-1}}\right)\right] \right\} \\ &\quad \text{s.t.} \sum_{s=1}^{S} x_{sj} \le W_j \\ &\quad 0 \le x_{sj} \le C_{sj},\ x_{sj} \text{ integer},\ s = 1, \ldots, S \end{aligned}$$

where N_j^d, N_{sj}^b, W_{j-1}, $B_{s,j-1}$, and $C_{s,j-1}$ are defined above.

The two boundaries of the formulation occur on day N and on day 0. On day N the system is initialized to empty ($W_N = 0$, $B_{sN} = 0$, and $C_{sN} = C$). The boundary conditions on day 0 arise from utilization costs incurred on the day of surgery. The schedule must be completed on the morning of day 0 and as a result, as many cases are taken off the RQ as possible on the morning of surgery. Deferral and blocking penalties are no longer an issue

because we assume that all elective demand arrives to the system prior to day 0's decisions. As with the daily deferral costs, the objective function on day 0 only penalizes those cases left on the RQ that could have been feasibly added to the schedule. This leads to the following value function for the boundary:

$$V_0(W_0, \boldsymbol{B_0}, \boldsymbol{C_0}) = \min_{\boldsymbol{x_0}} \left\{ h_0 \left(\min \left(\sum_{s=1}^{S} C_{s0}, W_0 \right) - \sum_{s=1}^{S} x_{s0} \right) + \sum_{s=1}^{S} r_{s0}(C_{s0} - x_{s0}) \right\}$$

$$\text{s.t.} \sum_{s=1}^{S} x_{s0} \leq W_0$$

$$0 \leq x_{s0} \leq C_{s0},\ x_{s0} \text{ integer}, s = 1, \ldots, S$$

2.4. *Sample path for a single OR example*

In order to better understand how the optimal decisions generated by the SDP translate to system costs, it is helpful to look at a sample path for a small numerical example. Suppose a single OR has capacity for four cases, and that the daily demand arrivals for both the primary specialty and RQ follow Poisson distributions. The daily arrival rates and system costs are specified in Table 1. The resulting SDP is solved to get the optimal decision for each feasible state on each day. Table 2 illustrates the sample path generated by single realizations of the arrival random variables, along with the optimal decisions and transitions generated by the SDP.

Of particular interest in this example are days 3 and 2, where the SDP solution calls for scheduling one case off the RQ for the given states. On day 3, the RQ consists of two cases but only one is taken, leading to a deferral penalty. Because there is still sufficient space in the room for that day's primary arrival ($T_3 < C_3 - x_3$), no blocking penalty is incurred and one blocking eligible case is passed to the following day. On day 2, another deferral penalty is incurred, and because the primary arrivals exceed the remaining available space ($T_2 > C_2 - x_2$) a blocking penalty is also incurred.

Table 1. Input data for simple SDSSP1 instance

Day (j)	4	3	2	1	0
Arrival Rates					
Primary Demand (E[T_j])	1	2	0.5	0.5	0
Secondary Demand (E[R_j])	1	1	1	1	0
Costs					
Deferral (h_j)	1	1	1	1	1
Blocking (r_j)	3	3	3	3	5

Note: OR Capacity (C) = 4.

Table 2. Sample path for simple SDSSP1 instance

Day (j)	4	3	2	1	0
W_j	0	2	2	2	3
B_j	0	0	1	1	0
C_j	4	4	2	0	0
x_j	0	1	1	0	0
T_j	0	1	2	1	0
R_j	2	1	0	0	0
N_j^d	0	1	1	0	3
N_j^b	0	0	1	1	0
Daily Costs	0	1	4	3	0

In the exploration of optimal policies for SDSSP1 (the single room SDP), a striking trend emerges. For all input data satisfying certain realistic assumptions, the optimal policy for each day follows what we describe as a threshold policy. That is, for each day before surgery there is a specific amount of space preserved for future primary arrivals, and the optimal decision takes as many cases as necessary to reach the threshold. If this number of cases is not feasible, then the decision takes the system as close to this threshold as possible. Furthermore, the threshold is independent of the secondary (RQ) demand arrival process. For the example above, the observed thresholds were (2, 3, 1, 1, 0) for days 4, ... ,0. Looking at these thresholds for the states observed in the sample path in Table 2 sheds light on why the corresponding decisions are made (i.e., on day 3, 4 spaces are available and the threshold is 3, so 1 RQ case is selected). The next section presents an analytical proof that the optimal policies for SDSSP1 always demonstrate this threshold behavior. The proof leads to a constructive algorithm for finding the desired thresholds, and thus all optimal decisions, for any set of input data without solving the full SDP.

3. Analytical results

Before formally stating the claim that SDSSP1 always produces threshold policies, several quantities must be defined. In addition to the value function (which is defined in the formulation as a minimized quantity), it is necessary below to define the function (F_j) that the value function minimizes. In these definitions, it is important to note that the costs, transitions, and expected values depend on the current state, the decision variable, and the demand arrival random variables. This dependence will be stated more explicitly when necessary. Because we are considering the single-room version of the SDP ($S = 1$), the index for the ORs is dropped from the notation.

First for $j = 1, \ldots, N$, and then for day of surgery ($j = 0$):

$$F_j(W_j, B_j, C_j, x_j) = h_j N_j^d + r_j E\left[N_j^b\right] + \mathbf{E}\left[V_{j-1}\left(W_{j-1}, B_{j-1}, C_{j-1}\right)\right]$$
$$F_0(W_0, B_0, C_0, x_0) = h_0(W_0 - x_0) + r_0(C_0 - x_0)$$

This allows us to restate the value function for $j = 0, \ldots, N$:

$$V_j\left(W_j, B_j, C_j\right) = \min\left\{F_j\left(W_j, B_j, C_j, x_j\right) | x_j \in \left\{0, 1, \ldots, \leq \min\left(W_j, C_j\right)\right\}\right\}$$

This notation also allows us to introduce a notation for the optimal SDP decision corresponding to state $\left(W_j, B_j, C_j\right)$. We also define a finite difference function for the function $F_j\left(W_j, B_j, C_j, x_j\right)$ with respect to x_j, which we will use later in our analysis to show the function's convexity.

$$x_j\left(W_j, B_j, C_j\right) = \arg\min\left\{F_j\left(W_j, B_j, C_j, x_j\right) | x_j \in \left\{0, 1, \ldots, \leq \min\left(W_j, C_j\right)\right\}\right\}$$
$$\Delta F_j\left(W_j, B_j, C_j, x_j\right) = F_j\left(W_j, B_j, C_j, x_j + 1\right) - F_j\left(W_j, B_j, C_j, x_j\right)$$

3.1. *Transition observations*

In the course of an induction proof for the threshold policy, the nature of the costs and state transitions for certain adjacent states and decision values will be important. The relationships below for $j = 1, \ldots, N$ result from the transition equations presented with the SDP formulation in Section II. If dependence on the state variables, decision variables, or arrival random variables is not shown, then these values are held constant in the differences below. The notation $I\{\ldots\}$ represents an indicator random variable, and the relationships are grouped for clarity.

Group 1 :
$$N_j^b\left(x_j + 1\right) - N_j^b\left(x_j\right) = I\left\{T_j \geq C_j - x_j\right\}$$
$$W_{j-1}\left(x_j + 1\right) - W_{j-1}\left(x_j\right) = -I\left\{T_j < C_j - x_j\right\}$$
$$B_{j-1}\left(x_j + 1\right) - B_{j-1}\left(x_j\right) = I\left\{T_j < C_j - x_j\right\}$$
$$C_{j-1}\left(x_j + 1\right) - C_{j-1}\left(x_j\right) = -I\left\{T_j < C_j - x_j\right\}$$

Group 2 :
$$N_j^b\left(B_j + 1, C_j - 1\right) - N_j^b\left(B_j, C_j\right) = I\left\{T_j \geq C_j - x_j\right\}$$
$$W_{j-1}\left(W_j - 1, C_j - 1\right) - W_{j-1}\left(W_j, C_j\right) = -I\left\{T_j < C_j - x_j\right\}$$
$$B_{j-1}\left(B_j + 1, C_j - 1\right) - B_{j-1}\left(B_j, C_j\right) = I\left\{T_j < C_j - x_j\right\}$$
$$C_{j-1}\left(C_j - 1\right) - C_{j-1}\left(C_j\right) = -I\left\{T_j < C_j - x_j\right\}$$

Group 3 :
$$N_j^b\left(B_j + 1, C_j - 1, x_j - 1\right) = N_j^b\left(B_j, C_j, x_j\right)$$
$$W_{j-1}\left(W_j - 1, C_j - 1, x_j - 1\right) = W_{j-1}\left(W_j, C_j, x_j\right)$$
$$B_{j-1}\left(B_j + 1, C_j - 1, x_j - 1\right) = B_{j-1}\left(B_j, C_j, x_j\right)$$
$$C_{j-1}\left(C_j - 1, x_j - 1\right) = C_{j-1}\left(C_j, x_j\right)$$

The Group 1 differences show how the number of blocking penalties and the state transitions are related when one additional case is scheduled off the RQ, as occurs when computing the finite difference $\Delta F_j\left(W_j, B_j, C_j, x_j\right)$ defined above. Using the insight from these Group 1 observations, we see that the subsequent day $j - 1$ states for adjacent day j decisions will be similarly "adjacent" in the manner implied by the Group 2 and 3 differences. In the course of the proof of the theorem presented in the next subsection, these observations will allow us to explicitly manipulate the finite difference function and show the convexity of the function $F_j\left(W_j, B_j, C_j, x_j\right)$ that we are seeking to minimize.

3.2. *Structure of optimal SDSSP1 policies*

These relationships provide the necessary insights to proceed with a formal statement and proof of the threshold policy suggested above. Two assumptions on the input data are required: (1) $h_j \leq r_j, \forall j$ and (2) $r_{j+1} \geq r_j, \forall j \geq 1$. These do not limit the strength of the result, because (1) deferral costs are certain while blocking penalties depend on uncertain future arrivals and (2) increasing the blocking penalty as the day of surgery approaches would discourage filling up the remaining space. In the statement of the theorem, part (iii) is the desired threshold result, while the other parts are necessary in the development of a proof by weak induction.

Theorem: For $j = N, \ldots, 1$ and all feasible states $\left(W_j, B_j, C_j\right)$,

(i) $\exists$ a function $G_j(n)$ s.t. $G_j(n)$ is non-increasing in n, $G_j(1) \leq r_j$ and $\Delta F_j\left(W_j, B_j, C_j, x_j\right) = G_j\left(C_j - x_j\right)$
(ii) $F_j\left(W_j, B_j, C_j, x_j\right)$ is convex in x_j
(iii) $\exists K_j$ satisfying $G_j\left(K_j + 1\right) < 0 \leq G_j\left(K_j\right)$ s.t. $x_j\left(W_j, B_j, C_j\right) = \min\left(W_j, \max\left(0, C_j - K_j\right)\right)$
(iv) $V_j\left(W_j - 1, B_j + 1, C_j - 1\right) - V_j\left(W_j, B_j, C_j\right) = \begin{cases} 0 & \text{if } C_j - K_j > 0 \\ G_j\left(C_j\right) & \text{if } C_j - K_j \leq 0 \end{cases}$

Base Case ($j = 1$): Observe the following statements about day 0. First, no OR slots need to be preserved for future arrivals, leading to an effective threshold of $K_0 = 0$. This gives the desired structure to the optimal decisions.

$$x_0(W_0, B_0, C_0) = \min(W_0, \max(0, C_0 - K_0)) = \min(W_0, C_0)$$

Second, using this choice of x_0 the optimal day 0 value function, with some simplification, can be written in terms of $(W_0 - C_0)$. This observation leads to the following equality for day 0.

$$V_0(W_0 - 1, B_0 + 1, C_0 - 1) = V_0(W_0, B_0, C_0)$$

Using the Group 1 transition relationships and this day 0 equality, the first two desired statements for day 1 emerge:

$$\begin{aligned}
&\Delta F_1(W_1, B_1, C_1, x_1)\\
&= F_1(W_1, B_1, C_1, x_1+1) - F_1(W_1, B_1, C_1, x_1)\\
&= h_1\left[N_1^d(x_1+1) - N_1^d(x_1)\right]\\
&\quad + r_1\left[N_1^b(x_1+1) - N_1^b(x_1)\right]\\
&\quad + E\left[V_0(W_0(x_1+1), B_0(x_1+1), C_0(x_1+1))\right.\\
&\quad \left. - V_0(W_0(x_1), B_0(x_1), C_0(x_1))\right]\\
&= -h_1 + r_1 E[I\{T_1 \geq C_1 - x_1\}]\\
&= -h_1 + r_1 P[T_1 \geq C_1 - x_1]\\
&= G_1(C_1 - x_1)
\end{aligned}$$

where $G_1(n) = -h_1 + r_1 P[T_1 \geq n]$

Note that $G_1(1) \leq -h_1 + r_1 \leq r_1$. Also, $G_1(n)$ is clearly non-increasing in n, which gives $\Delta F_1(W_1, B_1, C_1, x_1)$ non-decreasing in x_1 and proves the convexity of $F_1(W_1, B_1, C_1, x_1)$ with respect to x_1.

Minimizing $F_1(W_1, B_1, C_1, x_1)$ then suggests looking for the point where the finite difference changes signs. In other words, seek out K_1 such that $G_1(K_1+1) < 0 \leq G_1(K_1)$ and try to set $x_1 = C_1 - K_1$. Such a K_1 exists because $G_1(0) = -h_1 + r_1 \geq 0$ (by assumption on input data) and because $G_1(n) \to -h_1 < 0$ as $n \to \infty$. If the desired value for x_1 is infeasible, then choose x_1 on the boundary closest to the desired value. This gives the desired expression for the optimal decision in terms of the threshold K_1.

$$x_1(W_1, B_1, C_1) = \min(W_1, \max(0, C_1 - K_1))$$

All that remains for the base case is to show the final piece of the claim, which is critical for the inductive step. For brevity, define $x_1^{**} = x_1(W_1 - 1, B_1 + 1, C_1 - 1)$ and $x_1^* = x_1(W_1, B_1, C_1)$. From the expression above for the optimal decisions, there are two cases to consider: (1) $x_1^{**} = x_1^* = 0$ when $C_1 - K_1 \leq 0$ and (2) $x_1^{**} = x_1^* - 1$ when $C_1 - K_1 > 0$.

Case 1 ($x_1^{**} = x_1^* = 0$): The group 2 transition relationships state that in this scenario the subsequent day 0 states (from the corresponding x_1^{**} and x_1^* states) will either be identical (for $T_1 \geq C_1$) or differ in such a way that by the day 0 observations above they will have the same value (for $T_1 < C_1$). The desired difference then depends only on the differences in deferral and blocking costs.

$$\begin{aligned}
&V_1(W_1 - 1, B_1 + 1, C_1 - 1) - V_1(W_1, B_1, C_1)\\
&\quad = h_1\left[N_1^d(W_1 - 1, C_1 - 1) - N_1^d(W_1, C_1)\right]\\
&\qquad + r_1 E\left[N_1^b(B_1 + 1, C_1 - 1) - N_1^b(B_1, C_1)\right]\\
&\quad = -h_1 + r_1 P[T_1 \geq C_1]\\
&\quad = G_1(C_1)
\end{aligned}$$

Case 2 ($x_1^{**} = x_1^* - 1$): According to the group 3 transition relationships, the blocking costs and subsequent day 0 states will be identical in this scenario. The deferral costs will also be the same, giving:

$$V_1(W_1 - 1, B_1 + 1, C_1 - 1) - V_1(W_1, B_1, C_1) = 0$$

These two cases yield the final piece of the base case.

Inductive Step: Assume that all parts of the claim hold for day $j-1$ and proceed with part (i) of the claim.

$$\begin{aligned}
&\Delta F_j(W_j, B_j, C_j, x_j)\\
&\quad = F_j(W_j, B_j, C_j, x_j + 1) - F_j(W_j, B_j, C_j, x_j)\\
&\quad = h_j\left[N_j^d(x_j + 1) - N_j^d(x_j)\right]\\
&\qquad + r_j E\left[N_j^b(x_j + 1) - N_j^b(x_j)\right]\\
&\qquad + E\left[V_{j-1}(W_{j-1}(x_j+1), B_{j-1}(x_j+1), C_{j-1}(x_j+1))\right.\\
&\qquad \left. - V_{j-1}(W_{j-1}(x_j), B_{j-1}(x_j), C_{j-1}(x_j))\right]
\end{aligned}$$

By the group 1 relationships, the day $j-1$states are identical when $T_j \geq C_j - x_j$. Otherwise, the states have the form of the difference in part (iv) of the induction assumption. By the induction assumption then, the resulting values are equal when $C_{j-1}(x_j) - K_{j-1} > 0$. But for $T_j < C_j - x_j$, it follows that $C_{j-1}(x_j) = C_j - x_j - T_j$. Combining these pieces with the induction assumption, the difference in day $j-1$ states is $G_{j-1}(C_j - x_j - T_j)$ when $C_j - x_j - K_{j-1} \leq T_j < C_j - x_j$, and is zero otherwise. This is reflected in the conditional expectation below, allowing the finite difference calculation to continue.

$$\begin{aligned}
&\Delta F_j(W_j, B_j, C_j, x_j)\\
&\quad = -h_j + r_j P\left[T_j \geq C_j - x_j\right]\\
&\qquad + E\left[G_{j-1}(C_j - x_j - T_j)\right.\\
&\qquad \left. | C_j - x_j - K_{j-1} \leq T_j < C_j - x_j\right]\\
&\quad = -h_j + r_j P\left[T_j \geq C_j - x_j\right]\\
&\qquad + \sum_{i=1}^{K_{j-1}} P\left[T_j = C_j - x_j - i\right] G_{j-1}(i)\\
&\quad = G_j(C_j - x_j)
\end{aligned}$$

where $G_j(n) = -h_j + r_j P[T_j \geq n] + \sum_{i=1}^{K_{j-1}} P[T_j = n - i] G_{j-1}(i)$

It is important to note that if $K_{j-1} = 0$, then the final summation in this expression disappears. Moving on to show that $G_j(n)$ possesses the desired qualities, the next two results use both the induction assumptions on $G_{j-1}(n)$ and the assumption that $r_j \geq r_{j-1}$.

$$\begin{aligned}
&G_j(1)\\
&\quad = -h_j + r_j P\left[T_j \geq 1\right] + \sum_{i=1}^{K_{j-1}} P\left[T_j = 1 - i\right] G_{j-1}(i)\\
&\quad = -h_j + r_j P\left[T_j \geq 1\right] + P\left[T_j = 0\right] G_{j-1}(1)\\
&\quad \leq -h_j + r_j P\left[T_j \geq 1\right] + P\left[T_j = 0\right] r_{j-1}\\
&\quad \leq -h_j + r_j\\
&\quad \leq r_j
\end{aligned}$$

$$
\begin{aligned}
&G_j(n-1) - G_j(n) \\
&= \left\{ -h_j + r_j P\left[T_j \geq n-1\right] \right. \\
&\quad \left. + \sum_{i=1}^{K_{j-1}} P\left[T_j = n-1-i\right] G_{j-1}(i) \right\} \\
&\quad - \left\{ -h_j + r_j P\left[T_j \geq n\right] + \sum_{i=1}^{K_{j-1}} P\left[T_j = n-i\right] G_{j-1}(i) \right\} \\
&= r_j P\left[T_j = n-1\right] + P\left[T_j = n-1-K_{j-1}\right] G_{j-1}\left(K_{j-1}\right) \\
&\quad + \sum_{i=1}^{K_{j-1}-1} P\left[T_j = n-i\right] \left(G_{j-1}(i) - G_{j-1}(i+1)\right) \\
&\quad - P\left[T_j = n-1\right] G_{j-1}(1)
\end{aligned}
$$

At this stage, note that $G_{j-1}\left(K_{j-1}\right) \geq 0$ by the selection of K_{j-1} and $G_{j-1}(i) - G_{j-1}(i+1) \geq 0$ and $G_{j-1}(1) \leq r_{j-1}$ by the induction assumption. Continuing with the computation gives:

$$
\begin{aligned}
G_j(n-1) - G_j(n) &\geq r_j P\left[T_j = n-1\right] - r_{j-1} P\left[T_j = n-1\right] \\
&\geq 0
\end{aligned}
$$

Therefore $G_j(n)$ is non-increasing in n, which gives $F_j\left(W_j, B_j, C_j, x_j\right)$ convex in x_j. Using the same argument presented in the base case, minimizing $F_j\left(W_j, B_j, C_j, x_j\right)$ requires finding K_j such that $G_j\left(K_j+1\right) < 0 \leq G_j\left(K_j\right)$ and trying to set $x_j = C_j - K_j$. Again, this K_j is guaranteed to exist because $G_j(0) = -h_j + r_j \geq 0$ and because $G_j(n) \to -h_j < 0$ as $n \to \infty$. If the desired value for x_j is infeasible, then choose x_j on the boundary closest to the desired value. This gives the required expression for the optimal decision in terms of the threshold K_j:

$$
x_j\left(W_j, B_j, C_j\right) = \min\left(W_j, \max\left(0, C_j - K_j\right)\right)
$$

To show part (iv) of the claim, define $x_j^{**} = x_j\left(W_j - 1, B_j + 1, C_j - 1\right)$ and $x_j^{*} = x_j\left(W_j, B_j, C_j\right)$. Just as in the base case, the expression for the optimal decisions yields two cases for the relationship between these two policies: (1) $x_j^{**} = x_j^{*} = 0$ when $C_j - K_j \leq 0$ and (2) $x_j^{**} = x_j^{*} - 1$ when $C_j - K_j > 0$.

Case 1 ($x_j^{**} = x_j^{*} = 0$): The group 2 transition relationships state that in this scenario the subsequent day $j-1$ states (from the corresponding x_j^{**} and x_j^{*} states) will either be identical (for $T_j \geq C_j$) or differ in such a way that part (iv) of the induction assumption may be applied (for $T_j < C_j$). Applying the induction assumption when $T_j < C_j$ gives that the difference in values between the day $j-1$ states will be $G_{j-1}\left(C_{j-1}\right)$ when $C_{j-1} - K_{j-1} \leq 0$, and will be zero otherwise. But when $T_j < C_j$, note that $C_{j-1}(0) = C_j - T_j$. This implies that the value of the day $j-1$ states will only be nonzero when $C_j - K_{j-1} \leq T_j < C_j$, a result which appears in the conditional expectation below.

$$
\begin{aligned}
&V_j\left(W_j - 1, B_j + 1, C_j - 1\right) - V_j\left(W_j, B_j, C_j\right) \\
&\quad = h_j\left[N_j^d\left(W_j - 1, C_j - 1\right) - N_j^d\left(W_j, C_j\right)\right] \\
&\qquad + r_j\left[N_j^b\left(B_j + 1, C_j - 1\right) - N_j^b\left(B_j, C_j\right)\right] \\
&\qquad + E\left[G_{j-1}\left(C_j - T_j\right) \middle| C_j - K_{j-1} \leq T_j < C_j\right] \\
&\quad = -h_j + r_j P\left[T_j \geq C_j\right] + \sum_{i=1}^{K_{j-1}} P\left[T_j = C_j - i\right] G_{j-1}(i) \\
&\quad = G_{j-1}\left(C_{j-1}\right)
\end{aligned}
$$

Case 2 ($x_j^{**} = x_j^{*} - 1$): As in the base case, the group 3 transition relationships show that the blocking costs and subsequent day $j-1$ states will be identical in this scenario. The deferral costs will also be the same, giving:

$$
V_j\left(W_j - 1, B_j + 1, C_j - 1\right) - V_j\left(W_j, B_j, C_j\right) = 0
$$

These cases complete the proof of the claim.

3.3. *SDSSP1 threshold algorithm*

The definition of $G_j(n)$ and part (iii) of the claim suggest a constructive algorithm to determine the optimal thresholds for any set of input data. It is clear that on the day of surgery the OR manager need not preserve any time for future primary demand, thus the algorithm begins by setting $K_0 = 0$. The statement of the theorem implies that the threshold for each day occurs where the finite difference function changes sign. Therefore, with the daily arrival distributions and deferral and blocking costs given as input, K_j can be found by iterating through $G_j(n)$ for $n = 0, 1, 2, \ldots$ until it changes signs. We note that for $j > 1$, $G_j(n)$ depends on $G_{j-1}(n)$ for $n = 0, 1, \ldots, K_{j-1}$. Therefore, the algorithm begins by finding the threshold for day 1, and then steps back iteratively to find the thresholds for earlier days. This process is stated explicitly in the following pseudocode, using the functions $G_j(n)$ defined in the proof of the theorem.

Optimal Thresholds Algorithm

(0) **Input**: h_j, r_j, and probability distributions for T_j for $j = 1, \ldots N$.
Initialize $K_0 = 0$.
(1) For $j = 1, \ldots N$:

Compute $G_j(i)$ for $i = 1, 2, \ldots$ until $G_j(i) < 0$.
Set $K_j = i - 1$ for the smallest i s.t. $G_j(i) < 0$.
Store $G_j(i)$ for $i = 0, 1, \ldots, K_j$ for use in computing K_{j+1}.

(2) **Output**: Thresholds K_j for $j = 0, \ldots, N$.

The power of this algorithm lies in the fact that it allows us to find the optimal solution for any state in SDSSP1 without solving the SDP by the computationally intensive, iterative method implied by Bellman's optimality equations.

Furthermore, the algorithm only requires input about the daily blocking and deferral costs and the distributions of the primary arrivals, directly informing the types of data and analyses that would be required in order to implement a threshold-based policy in a real-world OR suite. Most importantly, however, the optimality of the thresholds suggests a new way to think about block release dates and the more general process of releasing unused OR time to cases on the RQ. We will discuss this new perspective and its policy implications in the next section. In order to demonstrate the sensitivity of the optimal thresholds, and thus the optimal RQ decisions, to the input data, the next section also presents the results of a brief computational study.

4. Discussion and computational results

4.1. *Block release thresholds*

Recall that traditional block release dates specify a particular day before the day of surgery prior to which the OR manager may not release any open OR time to RQ cases and after which the manager may schedule RQ cases as he sees fit. It is interesting to observe that this type of block release policy can be viewed in terms of the thresholds generated by the solutions to SDSSP1. Specifically, a strictly enforced block release date implies a threshold equal to the capacity of the OR prior to the block release date (thus allowing no RQ cases to be scheduled) and a threshold equal to zero after the block release date (thus allowing as many RQ cases as possible to be scheduled). In contrast, the optimal thresholds produced by our model for SDSSP1 release unused OR time gradually over the course of several days leading up to the day of surgery. Figure 3 presents the contrast between the all-or-nothing block release implied by block release dates and the gradual block release implied by our threshold policies for a hypothetical instance of SDSSP1 with an OR capacity of eight hours. Viewed in this light, our results suggest that block release policies should be redesigned around optimal block release thresholds rather than remain focused on the block release dates currently used in practice.

An additional benefit of block release thresholds is their ability to adapt to a range of factors related to the development of a single day's OR schedule. In an OR suite with a large number of surgical specialties, traditional block release dates have only a limited ability to respond to differences in demand patterns and priority levels between the specialties. In practice, block release dates tend to range from one to five days before the day of surgery, reflecting only five possible policy settings. In contrast, the algorithm for our proposed block release thresholds directly incorporates and responds to these factors in the process of finding a set of thresholds and thus gives access to a much broader set of policies. The computational study presented in the next subsection demonstrates the sensitivity of the optimal thresholds to differences in arrival patterns and in blocking-to-deferral cost ratios (reflective of relative prioritization between primary and secondary demand). In spite of the added flexibility and more sophisticated policies suggested by our approach, the optimal thresholds themselves are still intuitive and easy to communicate and maintain the transparency of traditional block release dates.

4.2. *Sensitivity to input data*

In the algorithm for computing the optimal policy thresholds, it is interesting to note that the day 0 utilization costs play no role in determining the thresholds. Similarly, the room capacity plays no role in the algorithm and thus only impacts the thresholds indirectly through its more long-term impact on surgical demand. In fact, what drives the thresholds are the primary arrival distributions (T_j) and the ratio of blocking (r_j) to deferral (h_j) costs. To explore the sensitivity of the thresholds to this ratio, we test a range of ratios on different primary demand arrival scenarios. The goal of this analysis is to demonstrate the types of optimal thresholds the algorithm generates and

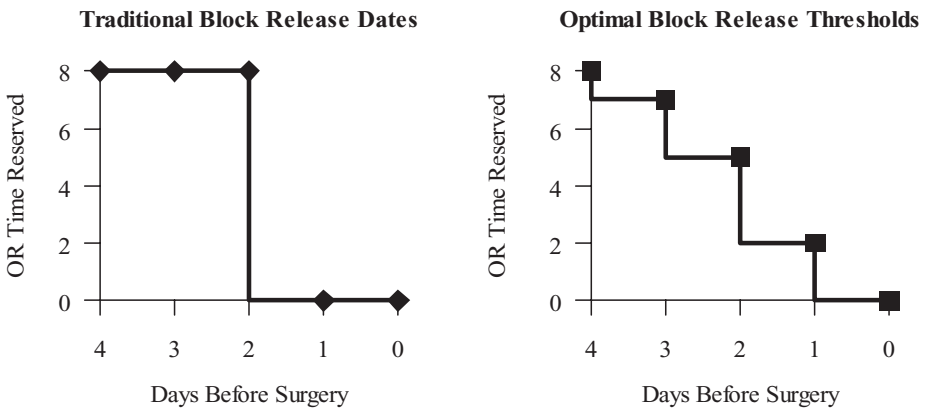

Fig. 3. Amount of OR time reserved for future primary arrivals by traditional block release dates and by the optimal block release thresholds from SDSSP1.

Table 3. Arrival rates $\left(E\left[T_j\right]\right)$ for 12 primary demand arrival scenarios

OR Capacity	C = 4					C = 8				
Days before surgery *(j)*	4	3	2	1	0	4	3	2	1	0
Arrival Pattern										
"Early" Demand										
$\sum_j E\left[T_j\right] > C$	2.5	1.5	0.5	0.5	0	5	3	1	1	0
$\sum_j E\left[T_j\right] = C$	2	1	0.5	0.5	0	4	2	1	1	0
$\sum_j E\left[T_j\right] < C$	1.5	0.5	0.5	0.5	0	3	1	1	1	0
"Late" Demand										
$\sum_j E\left[T_j\right] > C$	0.5	0.5	1.5	2.5	0	1	1	3	5	0
$\sum_j E\left[T_j\right] = C$	0.5	0.5	1	2	0	1	1	2	4	0
$\sum_j E\left[T_j\right] < C$	0.5	0.5	0.5	1.5	0	1	1	1	3	0

Note: Arrival random variables assumed to follow Poisson distributions.

how the threshold patterns respond to different input scenarios. The data requirements for a real-life scenario, and the challenges involved in collecting and estimating them, are discussed along with other future research plans in our concluding remarks.

As shown in Table 3, two OR capacities (of four and eight cases) were combined with arrival scenarios representing "early" and "late" arriving demand and total expected demand in excess of, equal to, and less than the OR capacity. We assume that each day's arrivals follow Poisson distributions with the rates shown in Table 3. The optimal SDSSP1 thresholds for these twelve arrival patterns were computed using the *Optimal Thresholds* algorithm for blocking-to-deferral cost ratios of 1:1, 3:1, 5:1, and 7:1 (with blocking and deferral costs remaining constant from day to day).

The optimal thresholds for each of the 32 test problems for each day before the day of surgery are presented in Table 4 (the thresholds are always 0 on the day of surgery). The threshold patterns discussed below for different arrival scenarios and increasing blocking-to-deferral cost ratios are consistent across both of the OR capacities studied, regardless of whether the total expected demand was greater than, equal to, or less than the stated capacity. The first trend to emerge is the observation that a blocking-to-deferral cost ratio of 1:1 leads to thresholds of 0 for all arrival patterns. This effectively greedy optimal policy (always scheduling as many RQ cases as possible) makes sense in light of the decision tree presented in Figure 1, where the deferral cost is certain and the blocking cost is dependent on random future arrivals.

For larger blocking-to-deferral ratios, the results show that the optimal thresholds adjust to the arrival patterns. For a ratio 3:1, the threshold for a given day roughly matches the expected number of arrivals for that day and day that follows (e.g., the threshold on day 4 is in line the number of arrivals expected on days 4 and 3). In contrast, the daily thresholds for the 5:1 ratio are more in line with the cumulative remaining expected arrivals, especially for the early demand scenarios. The daily thresholds for the 7:1 ratio are even more conservative, almost universally preserving more OR time than is required for the cumulative remaining expected arrivals.

When the primary demand is expected to arrival early, the thresholds start out fairly high on days 4 and 3 and quickly decrease as the day of surgery approaches. In contrast, when the primary demand is expected to arrive late, the thresholds are at their highest in the days just prior to surgery and in some cases actually increase as the day of surgery approaches. This reflects the fact that deferring a case for several days in a row can eventually be as costly as blocking a primary case. In these scenarios, if the secondary demand is high and arrives early, then a primary specialty with late-arriving demand runs the risk of losing its space before it has an opportunity to use it. One way for the OR stakeholders to avoid having a specialty lose its allocated time in this manner is to set a block release date (one key reason block release dates are used in practice). However, we note that a blocking cost structure with higher blocking costs before the demand arrival peak and lower costs after the peak could have the same effect by producing appropriate thresholds.

The common use of block release dates in practice, and the fact that they often differ between specialties, indicates some underlying, if unstated, complexities in the relative satisfaction costs that OR managers associate with deferring RQ decisions and blocking primary cases. Based on this analysis of SDSSP1, we argue for an approach that explicitly identifies the relative values placed on blocking and deferring and uses these values to set corresponding, threshold-based block release policies. In practice, these

Table 4. Optimal SDSSP1 decision thresholds for a range of arrival scenarios and blocking-to-deferral cost ratios

$r_j : h_j$	1:1	3:1				5:1				7:1			
Days before surgery (j)	$\forall_j$	4	3	2	1	4	3	2	1	4	3	2	1
$C = 4$													
"Early" Demand													
$\sum_j E[T_j] > C$	0	4	2	1	1	5	3	1	1	6	3	2	1
$\sum_j E[T_j] = C$	0	3	1	1	1	4	2	1	1	5	3	2	1
$\sum_j E[T_j] < C$	0	2	1	1	1	3	1	1	1	4	2	2	1
"Late" Demand													
$\sum_j E[T_j] > C$	0	1	2	3	3	3	4	4	4	5	5	5	4
$\sum_j E[T_j] = C$	0	1	2	2	2	3	3	3	3	4	4	4	4
$\sum_j E[T_j] < C$	0	1	1	2	3	3	3	3	4	4	4	4	4
$C = 8$													
"Early" Demand													
$\sum_j E[T_j] > C$	0	8	4	2	1	10	5	2	2	11	6	3	2
$\sum_j E[T_j] = C$	0	6	3	2	1	8	4	2	2	9	5	3	2
$\sum_j E[T_j] < C$	0	4	2	2	1	6	3	2	2	7	4	3	2
"Late" Demand													
$\sum_j E[T_j] > C$	0	3	5	7	6	8	8	9	7	9	9	10	7
$\sum_j E[T_j] = C$	0	3	4	5	5	6	6	7	6	7	7	7	6
$\sum_j E[T_j] < C$	0	2	3	3	4	4	4	4	4	6	5	5	5

values could be made to incorporate a range of practical concerns related to releasing blocks of OR time and scheduling RQ cases. For instance, deferral costs on the days immediately before surgery could reflect potential difficulties in getting last-minute RQ cases cleared for surgery (e.g., pre-operative testing or payment paperwork). Blocking costs could incorporate equipment requirements, room preferences associated with different primary specialties, and even differences in revenue contributions between specialties. Finding reliable methods for estimating these relative values, and determining how they differ from day-to-day and specialty-to-specialty, is an area ripe for future research.

5. Conclusion

The initial aim of this paper is to motivate and propose the dynamic single-day surgery scheduling problem as the first model in the literature to simultaneously capture the roles of block schedules, block release dates, and RQ decision-making policies in the dynamic evolution of a single day's OR schedule. To gain insight into more complex versions of the model, a simplified version with unit case durations is formulated as an SDP and a further simplified version with a single OR is analyzed. Our theoretical analysis reveals that optimal RQ decisions for SDSSP1 follow a threshold policy that preserves a desired amount of OR space on each day for yet-to-arrive primary demand. The proof of this result leads directly to an algorithm for computing these optimal thresholds. Our threshold-based approach to making daily RQ decisions leads to a fundamentally new perspective on designing optimal block release policies. Rather than release allocated blocks all at once, as dictated by traditional block release dates, our approach suggests releasing unused OR time gradually over the course of several days according to the optimal thresholds. The computational study demonstrates the sensitivity of our block release thresholds both to surgical demand arrival patterns and to the relative blocking and deferral costs.

Our modeling framework emerged from our discussions with collaborators in the peri-operative services department at UMMC, who have validated the importance of blocking and deferral costs in RQ decision-making, reviewed these results, and endorsed the concept of threshold-based policies. In order to develop specific threshold policies for this or any other facility, we first

must recall that SDSSP1 reflects a simplified version of a real-life surgery scheduling system. In practice, RQ decisions must be made with respect to multiple ORs and cases have a wide range of possible durations. Our continued research in this area has developed threshold-based heuristics for these more general settings, using target thresholds for each OR and then prioritizing between the ORs in an attempt to reach the desired thresholds (see Herring and Herrmann, 2011; Herring, 2011). Additional research is also required to show how the demand arrival distributions can be estimated from hospital surgical data and how the relative blocking and deferral costs can be estimated from a combination of data analysis and elicitation from various OR stakeholders. In order to further investigate the benefits of our proposed threshold-based block release policies, future research will also use data from UMMC to compare schedules developed using our thresholds with actual historical schedules on more traditional metrics such as OR utilization and throughput. In the interim, we believe that the results of this paper generate significant new insights into the surgery scheduling process and make a strong case for the use of threshold-based rules in the design of transparent and equitable block release and RQ policies.

Acknowledgments

We are indebted to the entire perioperative staff, and particularly Sheila Gilger, at the University of Maryland Medical Center in Baltimore for giving us access to their scheduling system and helping us to understand its intricacies. This project was supported by an OR of the Future grant (W81XWH-06-2-0057) from the US Army Medical Research Acquisition Activity.

References

Beliën, J. and Demeulemeester, E. (2007) Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research*, **176**, 1185–1204.

Beliën, J., Demeulemeester, E., and Cardoen, B. (2009) A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of Scheduling*, **12**(2), 147–161.

Blake, J. and Carter, M. (2002) A goal programming approach to strategic resource allocation in acute care hospitals. *European Journal of Operational Research*, **140**, 541–561.

Blake, J. and Donald, J. (2002) Mount Sinai Hospital uses integer programming to allocate operating room time. *Interfaces*, **32**(2), 63–73.

Brumelle, S. and Walczak, D. (2003) Dynamic airline revenue management with multiple semi-Markov demand. *Operations Research*, **51**, 137–148.

Cardoen, B., Demeulemeester, E., and Beliën, J. (2010) Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, **201**, 921–932.

Denton, B., Viapiano, J., and Vogl, A. (2007) Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, **10**, 13–24.

Dexter, F. and Macario, A. (2004) When to release allocated operating room time to increase operating room efficiency. *Anesthesia & Analgesia*, **98**, 758–762.

Dexter, F., Macario, A., and Traub, R. D. (1999) Which algorithm for scheduling elective add-on cases maximizes operating room utilization? Use of bin packing algorithms and fuzzy constraints in operating room management. *Anesthesiology*, **91**, 1491–1500.

Dexter, F. and Traub, R. (2002) How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia & Analgesia*, **94**, 933–942.

Dexter, F., Traub, R., and Macario, A. (2003) How to release allocated operating room time to increase efficiency: predicting which surgical service will have the most under-utilized operating room time. *Anesthesia & Analgesia*, **96**, 507–512.

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

Guinet, A. and Chaabane, S. (2003) Operating theatre planning. *International Journal of Production Economics*, **85**, 69–81.

Gupta, D. (2007) Surgical suites' operations management. *Production and Operations Management*, **16**(6), 689–700.

Hans, E., Wullink, G., van Houdenhoven, M., and Kazemier, G. (2008) Robust Surgery Loading. *European Journal of Operational Research*, **185**, 1038–1050.

Herring, W. (2011) Prioritizing patients: stochastic dynamic programming for surgery scheduling and mass casualty incident triage. Ph.D. Dissertation, University of Maryland, College Park, MD. Available at: http://hdl.handle.net/1903/11689.

Herring, W. and Herrmann, J. (2011) The single-day surgery scheduling problem: sequential decision-making and threshold-based heuristics. *OR Spectrum*. Available at: http://www.springerlink.com/content/X056175461528428/

Lee, T. and Hersh, M. (1993) A model for dynamic airline seat inventory control with multiple seat bookings. *Management Science*, **27**(3), 252–265.

Macario, A., Vitez, T., Dunn, B., and McDonald, T. (1995) Where are the costs in perioperative care? Analysis of hospital costs and charges for inpatient surgical care. *Anesthesiology*, **83**(6), 1138–1144.

McManus, M., Long, M., Cooper, A., Mandell, J., Berwick, D., Pagano, M., and Litvak, E. (2003) Variability in surgical caseload and access to intensive care services. *Anesthesiology.* **98**(6), 1491–1496.

Ozkarahan, I. (2000) Allocation of surgeries to operating rooms by goal programming. *Journal of Medical Systems*, **24**(6), 339–378.

Pham, D. and Klinkert, A. (2008) Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research*, **185**, 1011–1025.

Price, C., Golden, B., Harrington, M., Konewko, R., Wasil, E., and Herring, W. (2011) Reducing boarding in a post-anesthesia care unit. *Production and Operations Management*, **20**(3), 431–441.

Santibañez, P., Begen, M., and Atkins, D. (2007) Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a British Columbia health authority. *Health Care Management Science*, **10**, 269–282.

Schütz, H. and Kolisch, R. (2010a) Approximate dynamic programming for capacity allocation in the service industry (Working paper). Available at SSRN: http://ssrn.com/abstract=1618315.

Schütz, H. and Kolisch, R. (2010b) Capacity allocation for demand of different customer-product-combinations with cancellation, no-shows, and overbooking when there is sequential delivery of service (Working paper). Available at SSRN: http://ssrn.com/abstract=1618313.

Testi, A., Tanfani, E., and Torre, G. (2007) A three–phase approach for operating theatre schedules. *Health Care Management Science*, **10**, 163–172.

van Oostrum, J. M., van Houdenhoven, M., Hurink, J. L., Hans, E. W., Wullink, G., and Kazemier, G. (2008) A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum*, **30**(2), 355–374.