A STOCHASTIC DYNAMIC PROGRAM FOR THE SINGLE-DAY SURGERY SCHEDULING PROBLEM

William L. Herring, University of Maryland
Applied Mathematics & Statistics and Scientific Computation Program

Abstract

The scheduling of individual surgical patients has received extensive treatment in the literature, but this paper is the first to propose a model that simultaneously captures the roles of block schedules, block release dates, and request queue policies in the dynamic evolution of the schedule for a single day in an operating room suite. The model is formulated as a stochastic dynamic program and theoretical results are obtained for a single room version. These results demonstrate that optimal request queue decisions for a single room follow a threshold policy that preserves a desired amount of operating room space for the remaining demand from the room's allocated surgical specialty. An algorithm for determining the optimal thresholds is presented, followed by computational results.

I. 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. Scheduling problems have long captured the interest of researchers in other industries, so it comes as no surprise that the problem of scheduling surgical patients into operating rooms benefits from a robust literature. This interest in surgery scheduling is intensified by the key role that operating room (OR) scheduling plays in determining hospital occupancy levels and because the OR is the most resource-intensive and profitable unit of a hospital (Macario, Vitez, Dunn, & McDonald, 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, Demuelemeester, & Beliën (2010) for a thorough review). Like other scheduling problems, surgery scheduling approaches hope to make more efficient use of existing resources, but improved efficiency in the context of the health care system comes with the added societal benefit of increasing access to health care.

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, or service lines. 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 & Donald, 2002; Santibañez, Begen, & Atkins, 2007; Testi, Tanfani, & Torre, 2007). First stage decisions 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 & 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. In recent years, block scheduling models have transitioned from simply implementing desired allocation levels to leveling hospital bed occupancy and minimizing overcapacity (Beliën & Demeulemeester, 2007; Blake & Donald, 2002; 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, Viapiano, & Vogl, 2007; Guinet & Chaabane, 2003).

A fundamental, but understudied, element of the day-to-day job of surgery scheduling is the transition from the rigidity of the block schedule determined in the second stage to the flexibility required in the third stage to execute the schedule on a given day with a particular realization of the stochastic demand for surgery. To place the importance of this transition into the proper context, it is necessary to understand the dynamic way in which the schedule for a single day evolves.

Dynamics of Surgery Scheduling

For a given day, the OR time that has been allocated to specific surgical specialties is initially controlled exclusively by the service lines. That is, the specialties are free to choose and sequence patients within their allocated blocks as they see fit. In the meantime, specialties or surgeons that do not have allocated time submit their cases to the surgical request queue (RQ). In the period leading up to the day of surgery, OR managers try to accommodate these RQ cases by looking for unused space in rooms originally allocated to other specialties. In practice, there is a set day before surgery, referred to as the *block release* *date*, after which OR managers regain control of an allocated block. Before the block release date, RQ cases may not be placed into open space. After the block release date, the *request queue policy* determines how and when RQ cases will be placed into unused space. Together, the block release dates and request queue policies control the interaction between the block schedule and individual patient scheduling stages of surgery scheduling.[1]

The dynamic aspect of scheduling individual patients emphasizes the importance of considering the joint impact of block schedules, block release dates, and request queue policies when studying single-day surgery scheduling. However, no existing work in the surgery scheduling literature to date addresses all three of these components simultaneously. Some recent papers incorporate block schedules as constraints in individual patient scheduling models, but in each case the schedules are determined all at once, rather than evolved dynamically over time (Hans, Wullink, van Houdenhoven, & Kazemier, 2008; Pham & 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, Traub, & Macario, 2003; Dexter & 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 eliminating the role of the block release date (Dexter, Macario, & Traub, 1999; Dexter & Traub, 2002; Gerchak, Gupta, & Henig, 1996).

The remainder of this paper seeks to develop, analyze, and test a model for the dynamic single-day surgery scheduling problem that takes into account all three of the elements referenced above. 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 a single operating room instance of the SDP. The proof of these results suggests an algorithm for determining optimal decisions without fully evaluating the SDP. Section IV presents computational results on the sensitivity of the optimal decisions to the input data as well as a simulation study of different block release date and request queue policy combinations. Finally, Section V offers some concluding remarks and indicates directions for future research.

II. Problem Statement and Formulation

Problem Statement

Before formally stating the dynamic single-day surgery scheduling problem, it is important to understand the costs associated with the evolution of a schedule for a suite of operating rooms. These costs come in two forms: utilization costs related to under- and over-utilization of available OR space and customer satisfaction costs related to (1) leaving cases on the request queue for long periods of time and (2) blocking surgical specialties' access to their allocated OR space. The utilization costs are common in the literature, but the customer satisfaction costs are unique to this formulation and warrant further explanation.

Consider an OR schedule that several days before the day of surgery has space for one additional case and a RQ containing a single case that fits in the available space. The OR with open space has been allocated to a surgical specialty, referred to here as the primary service line. On this day, the OR manager is faced with the choice between placing the RQ case in the open space or deferring the decision to the following day. Deferring the decision has no immediate impact on utilization, yet is still undesirable because it forces the patient on the RQ and the surgeon associated with the case to wait another day for a decision. This customer satisfaction cost is defined as a *deferral cost*. However, the primary service line may still generate a case to fill the remaining space in its room. If the RQ case has been accepted, the OR manager runs the risk of blocking the primary service line's access to its allocated room. This satisfaction cost is defined as a *blocking cost*, and it is the balance between deferral costs and potential blocking costs that informs the manager's decision to accept or defer the RQ case. The stochastic decision tree in Figure 1 below illustrates the potential outcomes for this simple scenario.

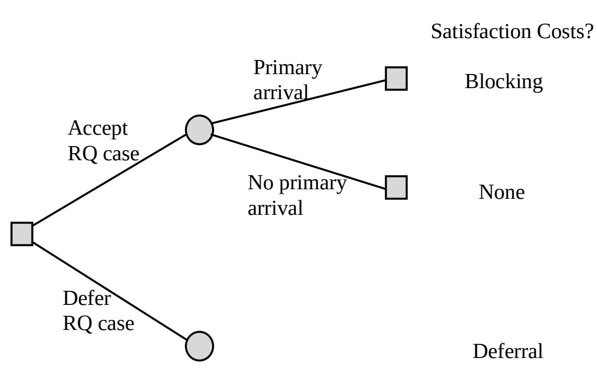

Figure 1. Decision tree for a simple request queue scenario

With the system costs defined as above, the dynamic single-day surgery scheduling problem can be stated as

[1] This description of a surgery scheduling system is based on a case study of the scheduling system at the University of Maryland Medical Center in Baltimore performed by the author in spring 2009. The existing work on block release policies in the literature (Dexter, Traub, & Macario, 2003; Dexter & Macario, 2004) suggests that this system can be generalized to other hospitals that use block scheduling.

follows. On a given day, a suite of operating rooms has been allocated to various surgical specialties, or primary service lines, according to a prescribed block schedule. These primary service lines generate demand for their blocks over several days before the day of surgery. Specialties that do not have an allocated block on this day may submit cases to the surgical request queue. The demand sources are assumed to be stochastic, both in quantity and in the timing of their arrival, and are independent of each other. Provided that its OR has open space, a primary service line may schedule its cases directly into the open space. Cases from the RQ can only be scheduled into open space by the OR manager, and if a block release date policy is in effect this may only be done after the block release date. If a primary service line generates a case that will not fit in its originally allocated space, this case is added to the RQ in hopes that it can be placed into another room with sufficient available space.

The OR manager's task is to choose on each day leading up to the day of surgery the number of RQ cases to add to each OR's schedule with the objective of minimizing the total system costs incurred in the process. A given day's decisions not only impact the costs encountered that day, but also impact the future state of the system and thus future costs. Block release dates serve as constraints restricting the days on which RQ cases can be added to the schedule, and may differ from room to room (or more likely, from specialty to specialty). Request queue policies dictate what decisions to make in what scenarios. The dynamic relationship between decisions and costs suggests a stochastic dynamic programming formulation. Because block release dates serve as constraints, it is clear that the optimal policy combination would use no block release dates and use the optimal values of the unconstrained SDP decision variables as the RQ policy. The formulation and analysis that follow consider an SDP unconstrained by block release dates. However, due to the practical arguments for using block release dates (protecting OR space allocated to primary service lines and reducing the number of future surgery days an OR manager must consider on any one day), the added cost of imposing block release dates will be explored in the computational results section.

Before describing the input data, states, and costs needed for the SDP formulation, it is important to review a few assumptions made in this analysis. First, it is assumed that RQ decisions are made once a day in the morning, before that day's demand for surgery has been generated. This is fairly realistic, because OR managers in practice are making decisions for several future surgery days at once and it is impractical to spend too much time on any one day. Also, demand for surgery is often communicated to the hospital late in the day after surgeons have completed time in a clinic or made their rounds.

A few other assumptions are made in order to simplify the initial formulation. The capacity of the operating rooms is assumed to be the same for all rooms and is stated in terms of the number of cases rather than hours. By extension, then, the RQ decisions reflect the number of cases to place into available space. The rooms are also assumed to be identical, allowing RQ cases to be scheduled into any OR. Priority levels are not explicitly associated with cases on the RQ, although once the number of cases to place is chosen it is easy to imagine an OR manager using implicit priorities to choose which cases. Finally, the swapping of scheduled cases between rooms and case cancellations are not considered, and the arrival of demand from a surgical specialty is assumed to be independent from day to day. Because of the novelty of this approach to surgery scheduling, these somewhat strong assumptions have been made in order to gain some initial insight into the behavior of the model. After this initial insight has been gained, the impact of these assumptions will be considered and relaxed versions of the model will be more readily approachable.

Stochastic Dynamic Programming 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
$C_N =$ capacity of a single OR

The stochastic demand for surgery is given by the following random variables. Arrivals are associated with their service lines, and a one-to-one correspondence between rooms and service lines 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. In the following definitions, $s = 1, \dots, S$ and $j = 0, \dots, N$.

$T_{sj} =$ arrivals to operating room s on day j generated by primary service line
$R_j =$ arrivals to the RQ on day j

The utilization and customer satisfaction costs are similarly defined by day and service line.

$h_0 =$ penalty for unscheduled cases left on RQ on the day of surgery
$r_{s0} =$ penalty for unused space in room s on the day of surgery

$h_j =$ deferral cost for day j
$r_{sj} =$ blocking cost for service line 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.

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

The number of blocking eligible cases in a room 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 service line'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.

x_{sj} = number of RQ cases to add to room 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.

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

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 service line'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.

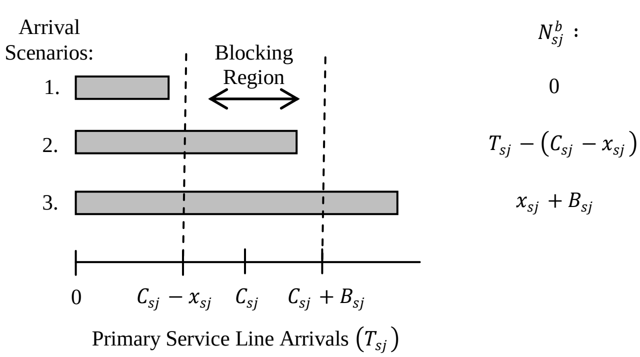

Figure 2. Illustration of different blocking penalty scenarios

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

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 service line arrivals that exceed the available space in their allocated room are redirected to the RQ.

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

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

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

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 $C_{sN} = C_N$). 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 demand for surgery is not generated on the day of surgery.[2] This leads to the following value function for the boundary:

$$V_o\left(W_0, \boldsymbol{B_0}, \boldsymbol{C_0}\right) = \min_{\boldsymbol{x_0}}\left\{h_0\left(W_0 - \sum_{s=1}^{S} x_{s0}\right) + \sum_{s=1}^{S} r_{s0}\left(C_{s0} - x_{s0}\right)\right\}$$
$$\text{s.t.} \quad \sum_{s=1}^{S} x_{s0} \le W_0$$
$$0 \le x_{s0} \le C_{s0}, \quad s = 1, \ldots, S$$

Single Room 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

[2] Realistically, urgent/emergent cases are often generated on the day of surgery, but these can be handled outside the proposed framework. In practice, entire blocks are reserved for urgent cases or stated room capacities are lowered to preserve slack for these cases. In either scenario, it is still desirable to minimize the costs associated with building the original schedule.

service line and RQ follow Poisson distributions. The daily arrival rates and system costs are specified below.

Day	4	3	2	1	0
Arrival Rates					
Primary Service	1	2	0.5	0.5	0
Request Queue	1	1	1	1	0
Costs					
Deferral	1	1	1	1	1
Blocking	3	3	3	3	5
Room Capacity = 4					

Table 1. Input data for simple SDP example

The resulting SDP is solved to get the optimal decision for each feasible state on each day. Consider the sample path generated by single realizations of the arrival random variables, combined with the optimal decisions and transitions generated by the SDP.

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	3

Table 2. Sample path for simple SDP example

Of particular interest in this example are days 3 and 2, where the SDP solution says to take 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 the primary service line 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 service line's arrivals exceed the remaining available space ($T_2 > C_2 - x_2$) a blocking penalty is also incurred.

In the exploration of optimal policies for the single room SDP, a striking trend emerged. For all input data satisfying certain realistic assumptions, the optimal policy for each day followed a kind of threshold policy. That is, for each day before surgery there was a specific amount of space that was preserved for future primary service line arrivals, and the optimal decision took as many cases as necessary to reach the threshold. If this number of cases was not feasible, then the decision took the system as close to this threshold as possible. Furthermore, the threshold was independent of the 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 were made (i.e. on day 3, 4 spaces were available and the threshold was 3, so 1 RQ case was selected). The next section presents an analytical proof that the optimal policies for the single room SDP 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.

III. Analytical Results

Before formally stating the claim that the single room SDP 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.

For $j = 1, \dots, N$:

$$F_j(W_j, B_j, C_j, x_j) = h_j \cdot N_j^d + r_j \cdot E[N_j^b] + E[V_{j-1}(W_{j-1}, B_{j-1}, C_{j-1})]$$

For $j = 0$:

$$F_0(W_0, B_0, C_0, x_0) = h_0(W_0 - x_0) + r_0(C_0 - x_0)$$

For all j:

$$V_j(W_j, B_j, C_j) = \min\{F_j(W_j, B_j, C_j, x_j)\} \quad \text{s.t.} \quad 0 \le x_j \le \min(W_j, C_j)$$

$$x_j(W_j, B_j, C_j) = \arg\min\{F_j(W_j, B_j, C_j, x_j)\} \quad \text{s.t.} \quad 0 \le x_j \le \min(W_j, C_j)$$

$$\Delta F_j(W_j, B_j, C_j, x_j) = F_j(W_j, B_j, C_j, x_j + 1) - F_j(W_j, B_j, C_j, x_j)$$

The final definition is a discrete form of a derivative which will be used in proving convexity. As with continuous functions, the convexity of a discrete function requires a non-decreasing derivative.

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.

<u>Group 1</u>

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

<u>Group 2</u>

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

<u>Group 3</u>

$$N_j^b(B_j+1, C_j-1, x_j-1) = N_j^b(B_j, C_j, x_j)$$
$$W_{j-1}(W_j-1, C_j-1, x_j-1) = W_{j-1}(W_j, C_j, x_j)$$
$$B_{j-1}(B_j+1, C_j-1, x_j-1) = B_{j-1}(B_j, C_j, x_j)$$
$$C_{j-1}(C_j-1, x_j-1) = C_{j-1}(C_j, x_j)$$

Structure of Optimal SDP 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) $h_j \leq r_j$ and (2) $r_{j+1} \geq r_j \quad \forall j \geq 1$. 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.

<u>Claim</u>: For $j = N, \ldots, 1$ and all feasible states (W_j, B_j, C_j)

(i) $\exists$ a function $G_j(n)$ s.t. $G_j(n)$ is non-increasing in n, $G_j(1) \leq r_j$, and $\Delta F_j(W_j, B_j, C_j, x_j) = G_j(C_j - x_j)$

(ii) $F_j(W_j, B_j, C_j, x_j)$ is convex in x_j

(iii) $\exists K_j$ satisfying $G_j(K_j+1) < 0 \leq G_j(K_j)$ s.t. $x_j(W_j, B_j, C_j) = \min(W_j, \max(0, C_j - K_j))$

(iv) $$V_j(W_j-1, B_j+1, C_j-1) - V_j(W_j, B_j, C_j) = \begin{cases} 0 & \text{if } C_j - K_j > 0 \\ G_j(C_j) & \text{if } C_j - K_j \leq 0 \end{cases}$$

Part (iii) is the desired threshold result, while the other parts are necessary in the development of the proof. The proof proceeds using weak induction.

<u>Base Case</u> ($j = 1$)

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 can be written in terms of $(W_0 - C_0)$, which gives the following equality.

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

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

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

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

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 derivative 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.

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

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.

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

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 derivative calculation to continue.

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

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

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 $r_j \geq r_{j-1}$.

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

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

At this stage, note that $G_{j-1}(K_{j-1}) \geq 0$ 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:

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

Therefore $G_j(n)$ is non-increasing in n, which gives $F_j(W_j, B_j, C_j, x_j)$ convex in x_j. Using the same argument presented in the base case, minimizing $F_j(W_j, B_j, C_j, x_j)$ requires finding K_j such that $G_j(K_j + 1) < 0 \leq G_j(K_j)$ 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.

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

In order to finish up the final part of the claim, define $x_j^{**} = x_j(W_j - 1, B_j + 1, C_j - 1)$ and $x_j^* = x_j(W_j, B_j, C_j)$. 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 $C_j - K_j \le 0$ 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 $T_j \ge C_j$) 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 $G_{j-1}(C_{j-1})$ when $C_{j-1} - K_{j-1} \le 0$, 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 $C_j - K_{j-1} \le T_j < C_j$, a result which appears in the conditional expectation below.

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

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:

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

These cases complete the proof of the claim. ■

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. The proof of the base case demonstrates that in addition to setting $K_0 = 0$, K_1 can be found by iterating through $G_1(n)$ for $n = 0, 1, 2,$ ... until it changes signs. Using K_{j-1} and storing $G_{j-1}(n)$ for $n = 1, 2, \ldots, K_{j-1}$, $G_j(n)$ can similarly be computed and K_j selected at the point where $G_j(n)$ changes sign.

Impact of Assumptions

Recall that this SDP formulation reflects a simplified version of the scheduling problem where room capacities and RQ decisions are stated in terms of the number of cases, essentially ignoring case durations. The nature of the threshold result for the single room SDP and its lack of limitations on the arrival distributions make the relaxation of this assumption quite feasible. Provided that the blocking and deferral costs are assumed to be proportional to case durations, the arrival distributions, state variables, and decision variables could easily be changed to reflect units of time. The resulting thresholds would then reflect the number of hours that the OR manager should preserve on each day for remaining arrivals. While the durations of the cases on the RQ might not allow the thresholds to be reached exactly, clever use of the $G_j(n)$ derivative values would still allow for effective decision-making.

IV. Computational Results

Sensitivity to Cost Ratios

In the algorithm for computing the optimal policy thresholds, it is interesting to note that the room capacity and day 0 utilization costs play no role in determining the thresholds. In fact, what drives the thresholds (aside from the arrival distributions) is the ratio of blocking to deferral costs. To demonstrate the sensitivity of the thresholds to this ratio, a range of ratios was tested on different arrival scenarios. Three arrival scenarios were considered, corresponding to early, middle, and late demand arrival patterns, and the daily arrivals were assumed to follow Poisson distributions with the rates shown in Table 3.

Arrival Scenarios	Days Before Surgery				
	4	3	2	1	0
Early	2	1	0.5	0.5	0
Middle	1	1	1	1	0
Late	0.5	0.5	1	2	0

Table 3. Primary service line arrival rates for three scenarios

For each of these scenarios, the day 0 costs were set to 0, the deferral costs were fixed at 1, and a range of blocking costs was selected. For simplicity, within each system the blocking cost remains constant from day to day. The optimal daily thresholds were computed using the algorithm suggested above for each combination of arrival pattern and blocking cost (Table 4). In each scenario, a blocking-to-deferral cost ratio of 1:1 leads to thresholds of 0, in effect equivalent to a greedy RQ policy. For larger ratios, the threshold patterns mirror the arrival patterns. Ratios in the range of 2:1 and 3:1 give day-to-day thresholds that roughly match the expected day-to-day arrivals, while ratios of 5:1 or higher yield thresholds that begin to mirror the cumulative remaining expected arrivals. For lower blocking costs, the thresholds in the late-arriving demand scenario start low and increase as the day of surgery approaches. This suggests that if RQ demand is high and arrives early, then a primary service line with late-arriving demand runs the risk of losing its space before it has an opportunity to fill it. If the OR stakeholders wish to avoid having a specialty lose its allocated space in this manner, then a blocking cost structure with higher blocking costs before the demand arrival peak and lower costs after the peak could be implemented to alleviate this tendency. Another alternative for avoiding this behavior is to set a block release date, which, while sub-optimal with regard to the SDP, is often more practical for OR managers due to the inherently relative nature of the blocking and deferral costs.

	Arrival Scenarios											
	Early				Middle				Late			
	Days Before Surgery											
Blocking Costs	4	3	2	1	4	3	2	1	4	3	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0
2	2	1	0	0	1	1	1	1	0	1	1	2
3	3	1	1	1	2	2	2	1	1	2	2	2
4	4	2	1	1	3	2	2	2	2	2	3	3
5	4	2	1	1	3	3	2	2	3	3	3	3
6	4	2	1	1	4	3	3	2	3	3	4	3
7	5	3	2	1	4	4	3	2	4	4	4	4

Table 4. Optimal SDP decision thresholds for a range of blocking costs across three demand arrival scenarios

Block Release Dates

Because of the practicality of block release dates, it is important to understand the extra costs incurred by implementing a block release policy. In order to explore these costs, a simulation environment was created for the development of the schedule for a single operating room. As above, the day 0 costs were set to 0 and the deferral costs were set to 1. The room capacity was set to 4 and the other input data is specified in Table 5, along with the corresponding optimal SDP thresholds. Because block release dates in practice range from a week to just a day before surgery, the arrival rates were extended beyond a week before surgery.

	Day Before Surgery						
	6	5	4	3	2	1	0
Blocking Costs	4	4	4	4	4	4	0
Primary Service Arrival Rates	0.25	1	1	1	0.25	0.5	0
Request Queue Arrival Rates	0.17	0.17	0.17	0.17	0.17	0.17	0
Optimal SDP Thresholds	2	2	2	2	1	1	0

Table 5. Input data and optimal SDP decision thresholds for testing the effects of block release dates

In addition to exploring the effects of block release dates, the simulation environment facilitates exploration of RQ policies other than the optimal thresholds generated by the SDP. One such alternative RQ policy is a greedy policy which takes as many cases as space allows (i.e. thresholds set to 0). Recall that a block release date is the first day on which RQ cases may be added to the schedule, and the RQ policy dictates how post-block release decisions should be made. For each policy combination shown in Table 6, the OR schedule was simulated over 1,000 replications. The optimal SDP solution is reflected by using the optimal thresholds and no block release date, and the costs of other scenarios are shown as percentage increases over this optimal combination.

Post-Block Release Date Policy	Block Release Date					
	None	4	3	2	1	0
Optimal Threshold	-	9.6	17.5	26.6	42.7	54.5
Greedy Threshold	17.1	31.0	39.9	31.8	50.1	54.5

Note: All apparent differences are statistically significant to 95% confidence.

Table 6. Percentage difference in average cost over optimal SDP thresholds with no block release date

The results in Table 6 show that the optimal thresholds significantly outperform the greedy thresholds for every possible block release date except for day 0 (when the two thresholds match). When the optimal thresholds are used, costs increase consistently as the block release dates move closer to the day of surgery. This makes sense intuitively, since reducing the block release date increases the number of days when the optimal decision is not allowed. In effect, lowering the block release date forces more deferrals when the optimal thresholds indicate that it would be preferable to run the risk of incurring blocking penalties. When the greedy thresholds are used, the costs also generally trend upward as the block release date is lowered. However, a dip is noticed when the block release date is day 2, corresponding to the day when the optimal threshold drops from 2 to 1. While the exact reason for this dip is difficult to ascertain, it does suggest that if a block release date must be used with greedy thresholds, a correct choice of block release date can effectively counteract some of the greedy behavior. Overall, these results suggest that if a block release date must be used for practical reasons, then it is important to combine it with the optimal thresholds.

V. 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 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 is formulated as an SDP and a single OR version is analyzed both theoretically and computationally. The theoretical analysis reveals that optimal request queue decisions for a single room should follow a threshold policy, where a desired amount of OR space is preserved on each day for yet-to-arrive surgical demand. The proof of this result leads directly to an algorithm for computing these optimal thresholds. The computational results demonstrate both the sensitivity of these thresholds to relative customer satisfaction costs and the added costs incurred by combining the thresholds with block release dates (as is often done in practice).

The insight gained from this analysis of the single OR problem suggests two primary directions for future research. The first involves the incorporation of case durations, as proposed at the end of Section III. The second involves trying to extend the threshold result to multiple room versions of the SDP, with and without case durations, where in the worst case the single room thresholds could be used as an efficient heuristic.

Acknowledgements

I would like to thank my advisor, Dr. Jeffrey Herrmann, for his patience and guidance throughout the course of this research. I am also indebted to the entire perioperative staff, and particularly Sheila Gilger, at the University of Maryland Medical Center for giving me access to their scheduling system and helping me 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. & Demeulemeester E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research, 176*, 1185-1204.

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

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

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

Dexter, F., Macario, A., & 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. & 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., & 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.

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

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

Guinet A. & 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., & Kazemier, G. (2008). Robust Surgery Loading. *European Journal of Operational Research*, *185*, 1038-1050.

Macario, A., Vitez, T., Dunn, B., & 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., & Litvak, E. (2003). Variability in surgical caseload and access to intensive care services. *Anesthesiology*. *98(6)*, 1491-1496.

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

Santibañez, P., Begen, M., & 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.

Testi, A., Tanfani, E., & 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., & Kazemier, G. (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum*, *30(2)*, 355-374.

Biographical Sketch

Will Herring earned his undergraduate degree in math at Davidson College and is currently a Ph.D. candidate in the Applied Math program at the University of Maryland College Park. His research focuses on the application of operations research models to the health care system. In addition to being a Student Member of IIE – Society for Health Systems, he is also a Student Member of the Institute for Operations Research and Management Sciences (INFORMS) and the American Mathematical Society (AMS). He was recently recognized with an Honorable Mention for the 2009 INFORMS Bonder Scholarship for Applied Operations Research in Health Services.