A Model of ICU Bumping

Gregory Dobson
Simon School of Business, University of Rochester, Rochester, New York 14627, dobson@simon.rochester.edu

Hsiao-Hui Lee
School of Business, University of Connecticut, Storrs, Connecticut 06269, hsiaohui.lee@business.uconn.edu

Edieal Pinker
Simon School of Business, University of Rochester, Rochester, New York 14627, ed.pinker@simon.rochester.edu

Many intensive care units (ICUs) face overcrowding. One response to this overcrowding is to bump ICU patients to other departments of the hospital to make room for new patient arrivals. Such bumping clearly has the potential to reduce quality of care. In this paper we develop a stochastic model of a single ICU with patient bumping. The purpose of this model is to enable planners to predict performance, in terms of bumping, under differing arrival patterns and capacity. We develop a Markov chain model and a new aggregation-disaggregation algorithm for this problem that enables us to keep track of the time in system for each patient despite the high dimensionality of the problem. Our approach allows for more accurate modeling of the system than previous work that assumed an exponential distribution for length of stay (LOS). We also demonstrate the superior computational efficiency of our approach over the Gauss-Seidel iterative method for solving the Markov chain. Finally, we use the model to explore how different surgery schedules influence bumping rates.

Subject classifications: health care; Markov chain; ICU; bumping; aggregation-disaggregation methods.
Area of review: Policy Modeling and Public Sector OR.
History: Received November 2008; revisions received June 2009, December 2009; accepted March 2010.

1. Introduction

Hospital intensive care units (ICUs) account for 15% to 20% of U.S. hospital costs (Gruenberg et al. 2006 and Pronovost et al. 2004). ICUs require expensive specialized resources, including both nurses and equipment. The demand for ICU care is high; for example, 90% of ICUs in New York state have insufficient capacity to provide proper care (Green 2002) and the occupancy rates are 90% (Pronovost et al. 2004).

Patients arrive at an ICU from a number of places. A patient could arrive from a scheduled surgery either because the nature of the surgery typically requires ICU care or because a routine procedure became more complex and required ICU care. The patient may arrive from the emergency department (ED) with or without a stop in the surgical suite. The patient may be directly admitted to the ICU by his/her physician, and finally, the patient may be a transfer patient from another hospital that does not have the facilities to handle the severity of the case. None of these sources results in a predictable arrival pattern at the ICU, although some sources such as scheduled surgery are more predictable in that the diagnosis is known before surgery begins and the history of similar cases can be used to predict the number of arrivals. ICUs can be specialized, and some specialties will experience more arrival predictability than others. ICU length of stay (LOS) is another source of uncertainty because the condition and prognosis of a patient when he arrives is variable.

ICU crowding has many negative effects on hospitals because it is a nexus for patient flows. New ICU patients cannot wait for a bed and must be served in some way. Empirically, Iapichino et al. (2004) show that higher occupancy rate leads to a higher mortality (lower quality of care). This result is not surprising because none of the standard actions taken at hospitals to deal with ICU overcrowding are appealing from the perspective of quality of care. If the patient is from the ED, the ICU can block him, thus causing the ED staff to continue to handle this severely sick patient until a bed in the ICU is free. This policy in turn may lead to ED overcrowding and/or diversions of ambulances to other EDs (Green 2002, McConnell et al. 2005). ICU crowding may also cause postponement or cancellation of surgeries in operating rooms (ORs) (Green 2002). If an arriving patient is from the OR, then the ICU can again block him, requiring the post anesthesia care unit (PACU) to administer care until an ICU bed is free. Finally, the ICU has the option to bump a patient who has been there for some days and is relatively healthier (Green 2002, Friedman and Steiner 1999). The ICU can place the bumped patient elsewhere: in a regular floor bed, a step-down unit, or another type of ICU (e.g., the Surgical ICU may bump patients to the Medical ICU or the Burn Trauma Unit). Similarly, new patients who are blocked may be sent directly to less-than-ideal hospital units. In each case either the intensity of care in terms of nursing-to-patient ratio is worse and/or the skill level of nursing is lower because they have less experience with this type of patient. In fact, Kc and

Terwiesch (2007) show that bumping decreases the patient length of stay (LOS) in ICUs but increases readmission rate. All the options described above result in patients being cared for in suboptimal facilities, thus lowering the average quality of care.

In this paper, we develop a stochastic model of a single ICU with patient bumping to create a planning tool that can be used to predict performance under differing arrival patterns and capacity. To simplify our study, we focus on bumping as a proxy for all the negative effects of crowding in ICUs. Given the high cost of ICU resources, the heavy loads, and the importance of the quality of care for vulnerable ICU patients, we believe a system performance model can be very valuable to planners. We calculate two performance measures of this bumping process as a function of the ICU's capacity, the arrival distribution, and the LOS distribution:

• probability that a randomly selected patient is bumped at sometime during their stay, and

• the expected number of days remaining for a patient who is bumped, and a comparison of these two measures across many scenarios.

Another important contribution is to provide a complete model for bumping, which requires keeping track of the status of every patient. Traditional hospital bed capacity planning papers, e.g., Green (2002), and de Véricourt and Jennings (2008), model these systems as an M/M/c queue, where every bed is assumed to be identical, the service time is assumed to be memoryless, and the performance measure is the service delay, and thus the system can be greatly simplified to a one-dimensional problem. When the interaction of bed dynamics and nursing workload is considered, a two-dimensional continuous-time Markov chain may be utilized to simplify the system if Markovian assumptions for arrivals and service time are made (Yankovic and Green 2007). However, in ICU operations, a complete system description requires knowing the status of every patient in the ICU, and the service process is not really memoryless. Therefore, it cannot be simplified to a one- or two-dimensional problem. In practice when a doctor needs to decide which ICU patient to bump she will not view all the ICU patients as identical but will attempt to pick out the healthiest one. This paper overcomes this limitation of previous work by explicitly tracking patient LOS. The result is a discrete time Markov chain (DTMC) model with a very large state space. We find that standard computational techniques for DTMC are not feasible for this problem, and therefore we develop a new specialized aggregation-disaggregation algorithm that can solve problems of practical size.

The outline of the rest of the paper is as follows. In §2 we review the relevant research. In §3 we introduce the model for the stationary arrival process, and in §4, because the model is too large to be solved even for moderately sized ICUs, we develop an exact aggregation-disaggregation method to solve the problem. In §5, we compare the computer resources needed by our procedure to that for the standard Gauss-Seidel iterative algorithm. In §6 we explain how we compute our performance measures. In §7 we do a numerical investigation to understand how OR schedules impact the ICU performance measures. In §8 we summarize and suggest ideas for further work.

2. Literature Review

Gruenberg et al. (2006, p. 502) say: "The cost of caring for patients in ICUs in the United States has been estimated to account for 1% to 2% of the gross national product and 15% to 20% of US hospital costs, which represents 38% of total US healthcare costs." Given its economic significance, ICU care has attracted much interest in the health-care literature. We can categorize these papers as either empirical or theoretical. Numerous empirical papers have looked at the mortality/length-of-stay distribution from hospital data. Knaus et al. (1993) study the clinic data for 42 ICUs in 40 hospitals, McManus et al. (2004) study an 18-bed medical-surgical ICU of a large, urban children's hospital during a two-year period, and Marik and Hedman (2000) study the different ways to quantify the LOS in ICU. We draw on these studies to generate parameter values for our numerical experiments in §6.

Numerous theoretical papers have studied hospital capacity planning and patient flow through an ICU (Bellandi and Rauber 1999, Green and Nguyen 2001, Green 2002, Ryan 2004, Harrison et al. 2005, Akcali et al. 2006). These papers typically employ either simulation or queueing theory. Bellandi and Rauber (1999) point out the fact that hospitals are running at their capacity in order to control operating expenses. However, due to the high variability and the increasing demand, a high utilization rate can be costly and hurt the hospitals. Green and Nguyen (2001) also identify this problem and discuss the impact of cutting hospital beds on patient service (in terms of bed delays). Using an M/M/c queueing model, Green (2002) examines the ICU unavailability data from New York State, concludes that 90% of the ICUs have insufficient capacity, and proposes more detailed models for capacity planning. Ryan (2004) provides a capacity expansion model with exponential demand growth and deterministic expansion lead time assumptions. Harrison et al. (2005) consider not only the mean but also the variability of the occupancy level. Furthermore, they also investigate the trade-off between the overflow frequency and the occupancy level for a fixed and a flexible bed allocation. Akcali et al. (2006) develop a network flow model to determine the optimal capacity under a budget constraint over a finite horizon.

For an ICU with sufficient capacity, incoming patients are almost always accommodated into the ICU. However, when an ICU is full, three policies may be applied to patients: bumping, misplacement, and rejection, where bumping gives absolute priority to a new patient, misplacement represents a patient being assigned to another or a

wrong ICU, and rejection gives priority to existing patients. Lowery (1992) finds that ICU admissions are generated either from scheduled surgeries or random arrivals and set up a simulation model with these two arrival sources. His goal is to investigate the effect of surgery workload increases. However, he does not discuss the bumping behavior in the ICU. In Lowery (1993), bumping behavior is incorporated, in a simulation model, in the following way: If a patient is eligible for bumping, the ICU bumps the patient out and accommodates the new patient. If no patient is eligible for bumping and other ICUs have excess capacity, the incoming patient is assigned to another ICU (misplacement). Otherwise, the patient is rejected for admission to the ICU and needs to wait for a bed in the PACU.

In the papers using queueing theory, the LOS is assumed to be exponential and hence memoryless, the arrival pattern is stationary through the week rather than periodic, and the focus is on the delay. De Bruin et al. (2007) use the Erlang loss model to incorporate rejection in a cardiac emergency setting. However, Kim et al. (1999) study a 14-bed ICU for six months and identify that the LOS for the patients from the scheduled surgeries is not exponential and the arrival rate from the scheduled surgeries is not Poisson. Both conclusions cast doubt on standard queueing model assumptions.

Our paper is different from the existing literature in that its main focus is to investigate the bumping in a single ICU caused by upstream patient flow using exact calculations rather than simulations. We allow for a nonstationary arrival pattern through the week and a LOS distribution that depends on the type of arrival, scheduled or unscheduled. We do not assume that LOS is memoryless and therefore track the current length of stay of each patient.

3. The Model

Although hospitals certainly run "24/7," the normal process of discharges and admissions follows a daily schedule. Elective surgeries are typically done during the day, and so arrivals to the ICU from this process occur from late morning to early evening. An ED sees the vast majority of its patients between noon and midnight, and so its contribution to admissions to the ICU would intermingle with that of the OR. Discharges that are planned for a given day tend to occur earlier in the day in anticipation of new arrivals. Although these events are certainly overlapping in reality, we model the ICU assuming that discharges for the day occur first and then admissions occur. Thus, we model time as discrete days and use a Markov chain.

For our purposes we characterize arrivals as two streams: scheduled and unscheduled. The scheduled patients arrive on only specified days of the week, e.g., when surgeons of the services that are handled by this ICU are scheduled, while unscheduled patients arrive on all days. The number arriving from each category is uncertain.

In addition to the uncertainty in the number of arrivals on a given day, the LOS for each patient is also uncertain,

further complicating the problem of utilizing these ICU resources effectively. The state of the system is defined by the remaining lengths of stay of the patients in the ICU. We first discharge those patients whose time in the ICU is completed. Next we consider the entire pool of patients who are requiring care from this particular ICU on that day. These include the current patients who have not been discharged and new patients who are arriving from either the scheduled or unscheduled sources. If there are enough beds for all these patients, then all the new patients are assigned beds. If the pool of patients exceeds the number of beds, then excess patients are bumped to other parts of the hospital. We assume the bumping is done based upon patient need. This completes the day.

The empirical research on ICU LOS shows that a large majority of patients have ICU stays of less than seven days but there are some outliers with very long stays. To capture this phenomenon while keeping the state space manageable, we define two types of patients: outliers (long LOS) and nonoutliers (short LOS). For a nonoutlier, we assume that when the patient arrives, he has a randomly generated LOS reflecting the clinically desired LOS but not the actual LOS in the ICU, as bumping may occur. Thus this clinically desired LOS is independent of the ICU capacity and other arrivals. For outliers the LOS will be at least as long as the longest nonoutlier but will evolve stochastically after arriving.

The capacity of the ICU is assumed to be a known number of beds C. By "bed" we mean the physical bed, the associated equipment (e.g., ventilator, monitors, etc.), and sufficient nursing staff to handle a patient in this bed 24 hours per day. Although it is possible to use overtime for nurses to expand ICU capacity (nursing typically being the bottleneck resource in ICUs), we assume that C incorporates the nursing overtime to which the hospital is willing to commit on a long-term basis.

On each day, indexed by i, N_i new patients arrive and are admitted to the ICU. We assume that $N_i \leqslant M$, the maximum number of arrivals per day. If $N_i = n$ patients arrive, then let L_j be the (random) LOS of patient j, $j = 1, \ldots, n$. $L_j \leqslant D$, where D is the maximum LOS. We define a patient j with $L_j = D$ as an outlier, while one with $L_j < D$ as a nonoutlier. We denote by the vector $\mathbf{a} = (a_1, \ldots, a_n)$ the ideal lengths of stay for these admitted patients, ordered so $a_1 \geqslant \cdots \geqslant a_n$. Thus the probability that admitted patients on a given day are described by the vector $\mathbf{a} = (a_1, \ldots, a_n)$ is

$$p(\mathbf{a}) \equiv c(\mathbf{a}) P\{N_i = n\} \prod_{j=1}^{n} P\{L_j = a_j\},$$

where $c(\mathbf{a})$ is the number of unordered vectors with the same components as $\mathbf{a}$. (Table 1 contains a summary of all the notation used in the paper.) We define A to be the set of all possible ordered arrival vectors. We assume that the arrival distribution is state independent. In practice, when

Table 1. Table of notation.

C	Number of beds (capacity)
D	Maximum length of stay
M	Maximum number of arrivals
N_i	Number of new arrivals on day i
S	Set of states
$\mathbf{s}, \mathbf{t}$	States
S_r	Set of states of rank r
S^l	Set of states of level l, $l = 0, \ldots, C - 1$
$\mathbf{x}, \mathbf{y}$	Aggregated states of level l, $l = 0, \ldots, C - 1$
S_r^l	Set of states of rank r and level l, $l = 0, \ldots, C - 1$
$S(\mathbf{v})$	The set of states that begin with vector $\mathbf{v}$
	$S_r(\mathbf{v})$, $S^l(\mathbf{v})$, $S_r^l(\mathbf{v})$ are defined analogously.
A	Set of all possible (ordered) arrival patterns
$\mathbf{a}$	$= (a_1, \ldots, a_n)$ an arrival pattern
$A_{(s,u)t}$	Set of arrival patterns that will allow the chain to transition from state $\mathbf{s}$ to state $\mathbf{t}$ conditioned on u outliers staying as outliers
$A_{(x,u)y}^l$	Set of arrival patterns that will allow the chain to transition from state x of level l to state $\mathbf{y}$ conditioned on u outliers staying as outliers of level l, $l = 0, \ldots, C - 1$
π	Stationary probability for all states $\mathbf{s}$
π^l	Stationary probability for all states of level l, S^l
$r(\mathbf{x})$	The rank of a state $\mathbf{x}$
$l(\mathbf{x})$	The level of a state $\mathbf{x}$
$d(\mathbf{x})$	The dimension of state $\mathbf{x}$, equal to the length of the vector
$o(\mathbf{x})$	The number of outliers of state $\mathbf{x}$
$p(\mathbf{a})$	The probability of an arrival pattern $\mathbf{a}$
P_{st}	The probability of a transition from $\mathbf{s}$ to $\mathbf{t}$
P_{xy}^l	The probability of a transition from state $\mathbf{x}$ to $\mathbf{y}$ in a chain of level l, $l = 0, \ldots, C - 1$
P_B	The probability of bumping someone on a given day
R	The average days remaining for a bumped patient
ρ	The utilization of the beds

an ICU is full, new admissions to the hospital might be diverted, thus reducing the arrival rate to the ICU. This blocking or demand reduction is beyond the scope of this paper.

Finally, we describe the bumping policy we model. We assume that if it is necessary to bump k patients, we remove the k patients with the least remaining days left in their stay (least time remaining bumped first). If two patients have the same number of days left and only one must leave, we pick arbitrarily. When bumping, we do not distinguish between newly arriving or existing patients; rather, we distinguish patients by their remaining LOS. We are assuming that the days remaining is a proxy for the health of the patient and that it is preferable to bump healthier patients rather than sicker ones. Note that we do not assume that the doctors making the bumping decisions know exactly the remaining LOS of each patient but rather they are capable of rank ordering the patients in terms of remaining LOS. There is empirical evidence that ICU doctors are capable of such assessments (see Vicente et al. 2004, Tu and Mazer 1996).

The state of the system is defined by a C-vector indicating the remaining days for the patient in each bed. If a bed is empty, then its remaining days is equal to 0. The state

space is large. For example, a moderately sized ICU of $C = 15$ beds with a maximum LOS of $D = 6$, has $7^{15} = 4.7 \cdot 10^{12}$ states. Fortunately, the identity of each bed is not important but rather just the days remaining, and by sorting the components of the state vector we can collapse the number of states. For our example with $C = 15$ and $D = 6$ we reduce the number of states to 54,264. This number is still too large for computing the stationary distribution of the associated Markov chain, and so in §4 we present an exact aggregation-disaggregation method that greatly reduces the computations necessary to determine the stationary distribution of the Markov chain. For the remainder of the paper we will work with the ordered state space.

DEFINITION 1. Let S be the set of states, namely

$$S = \left\{ \mathbf{s} = (s_1, \ldots, s_C) \in \mathbb{Z}^C \mid D \geqslant s_1 \geqslant s_2 \geqslant \cdots \geqslant s_C \geqslant 0 \right\},$$

and let A be the set of arrival vectors, namely,

$$A = \left\{ \mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n \mid n = 1, \ldots, M, \right.$$
$$\left. D \geqslant a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n > 0 \right\}.$$

We define p^O as the probability that an outlier patient remains an outlier the next day. This enables us to model the right tail of the LOS distribution as a geometric process. Define $o(\mathbf{s})$ as the number of outliers of state $\mathbf{s}$. Then the the probability of u outliers out of $o(\mathbf{s})$ staying as outliers can be computed as $P_o(u, o(\mathbf{s})) = \binom{o(\mathbf{s})}{u}(p^O)^u(1 - p^O)^{o(\mathbf{s})-u}$. If $u > o(\mathbf{s})$, then $P_o(u, o(\mathbf{s})) = 0$, and if $o(\mathbf{s}) = 0$, then $P_o(0, 0) = 1$. Thus by conditioning on the number of outliers who will stay as outliers, we can compute the state in the beginning of the next period (before the new arrivals). Given an arrival $\mathbf{a}$, a state $\mathbf{s}$, and u outliers staying as outliers, we can determine, based on the bumping rules mentioned previously, the state to which the chain will move. First we define an operation on two vectors and u. Let $f_k(\mathbf{a}, (\mathbf{s}, u))$ be a k-vector obtained from concatenating the vector $\mathbf{a} = (a_1, \ldots, a_n)$ of length n and the vector $(s_1, s_2, \ldots, s_u, (s_{u+1} - 1)^+, \ldots, (s_C - 1)^+)$ of length C into a single vector of length $n + C$, then sorting it in decreasing order and finally truncating it by only taking the first k components.

Given an ICU in state $\mathbf{s}$ which in a given day experiences an arrival pattern $\mathbf{a}$ and u outliers remaining as outliers, the next state of the system will be $f_C(\mathbf{a}, (\mathbf{s}, u))$. To see this, observe that first, as we move to the next day, the remaining stay for each person currently in the ICU will decrease by 1 (if not staying as an outlier) and any empty bed, whose corresponding remaining days is 0, will still be 0. Thus the ith component of $\mathbf{s}$ becomes $(s_i - 1)^+$ for $i > u$. Next the new patients arrive with their corresponding lengths of stay described by $\mathbf{a}$. We create a combined set of patients by concatenating the two vectors. The components are sorted in decreasing order so that we rank them in order of their need for the ICU. Next we remove any patients

who cannot fit in the C beds (components of the new vector past position C), because there are only C beds.

It is now possible to define the Markov chain's transition matrix, $\mathbf{P}$. Let $\mathbf{s}, \mathbf{t} \in S$ and define $A_{(s,u)t} = \{\mathbf{a} \in A \mid f_C(\mathbf{a}, (s, u)) = \mathbf{t}\}$. The probability of making a transition from $\mathbf{s}$ to $\mathbf{t}$ is

$$P_{st} = \sum_{u=0}^{o(s)} P_o(u, o(\mathbf{s})) \sum_{a \in A_{(s,u)t}} p(\mathbf{a}).$$

Let π be the stationary probability vector, namely the solution to $\pi = \pi \mathbf{P}$ and $\sum_{s \in S} \pi_s = 1$.

PROPOSITION 1. *The stationary probability vector π exists and is unique.*

PROOF. Let Q be the set of states that can be reached from the empty system (state $\mathbf{0} = (0, 0, \ldots, 0)$) by some sequence of arrival vectors, including the arrival vector $\mathbf{0}$ in some periods. Not all the states in S are in Q, but our choice of the state space S is notationally and computationally convenient, so it includes some transient states as the following argument makes clear. The probability of zero arrivals, $p(\mathbf{0})$, is positive, so every state $\mathbf{s} = (s_1, \ldots, s_C) \in S$ can transition in s_1 steps to state $\mathbf{0}$ with a probability of $P_o(0, o(\mathbf{s}))(p(\mathbf{0}))^{s_1} > 0$. Because the matrix is finite, we need to show that the states in U all communicate with one state, state $\mathbf{0}$. This is clear because by definition of $s \in Q$, we can reach s from $\mathbf{0}$, and from the argument above we can get back to $\mathbf{0}$; thus, Q is a closed irreducible set. This fact ensures that there is a unique $\{\pi_s\}_{s \in Q}$. The remaining states are transient so $\pi_s = 0$ for $s \in S - Q$. □

Periodic Schedules

We are interested in studying the effect of day-of-the-week seasonality on ICU operations. Therefore we need to be able to handle a weekly periodic arrival pattern. If the distribution of arrivals varies according to a weekly pattern, then it is possible to model that situation by expanding the state space. For each state $\mathbf{s}$ we create new states $(i, \mathbf{s})$ $i = 0., \ldots, 6$ for the seven days of the week, and for each possible transition from $\mathbf{s}$ to $\mathbf{t}$ we create a transition from $(i, \mathbf{s})$ to $(i+1 \mod 7, \mathbf{t})$. If $p_i(\mathbf{a})$ is the probability of observing an arrival $\mathbf{a}$ on day i of the week, then the transition matrix for the larger Markov chain is

$$P_{(i,s)(i+1 \mod 7, t)} = \sum_{a \in A_{st}} p_{i+1}(\mathbf{a}).$$

This transition probability can be extended to include outliers as well. The new chain has seven times the number of states and the aggregation-disaggregation procedure we describe in the next section works identically on this larger chain.

4. The Aggregation-Disaggregation Technique

The state space S is too large even for moderately sized ICUs, so we now investigate an aggregation-disaggregation process that will allow us to compute the stationary probability distribution with substantially less computing resources. As a first step to defining the set of aggregated states, we partition the state space S into $\{S_r\}_{r=1}^C$, where we call S_r the set of states of rank r which is defined as follows:

$$S_r = \{\mathbf{s} \in S \mid s_1 \geqslant s_2 \geqslant \cdots \geqslant s_r \geqslant r \geqslant s_{r+1} \geqslant \cdots \geqslant s_C\},$$
$$\text{for } r = 1, \ldots, C.$$

For example, for a six-bed ICU, the state $(11, 10, 6, 5, 4, 3)$ belongs to S_4 but the state $(11, 10, 6, 5, 5, 3)$ belongs to S_5. Note that if $D < C$, then $S_r = \varnothing$ for $r = D+1, \ldots, C$.

PROPOSITION 2. *The collection $\{S_r\}_{r=1}^C$ is a partition of S.*

PROOF. Let $r_1 < r_2$ and observe that if $\mathbf{s}^1 \in S_{r_1}$ and $\mathbf{s}^2 \in S_{r_2}$, then

$$s_{r_1+1}^2 \geqslant s_{r_2}^2 \geqslant r_2 > r_1 \geqslant s_{r_1+1}^1 \geqslant s_{r_2}^1, \tag{1}$$

where we chose $r_2 > r_1$ so $r_2 \geqslant r_1 + 1$ which in turn gives us the first and fifth inequalities because the components of the state are ordered, and the second and fourth follow from the fact that $\mathbf{s}^2 \in S_{r_2}$ and $\mathbf{s}^1 \in S_{r_1}$. Now (1) implies that $r_2 > s_{r_2}^1$, thus $\mathbf{s}^1 \notin S_{r_2}$, and that $s_{r_1+1}^2 > r_1$, thus $\mathbf{s}^2 \notin S_{r_1}$ and $S_{r_1} \cap S_{r_2} = \phi$. We define $S_0 = \{\mathbf{0}\}$. The state vector $\mathbf{0}$ is an empty ICU. Now take any $\mathbf{s} \in S$ with $\mathbf{s} \neq \mathbf{0}$. Start with $r = 1$, test if $\mathbf{s} \in S_r$. Clearly $s_1 \geqslant 1$. If the second condition $r \geqslant s_{r+1}$ fails, then repeatedly increase r by 1 and test $\mathbf{s} \in S_r$ until either it passes the test or $r = C$. The test must succeed, which demonstrates that $\mathbf{s} \in S_r$ for some r and thus $\{S_r\}_{r=1}^C$ is a partition of S. We conclude that if $\mathbf{s}^1 \in S_{r_1}$, then $\mathbf{s}^1 \notin S_{r_2}$ for any $r_1 \neq r_2$ and therefore the rank is unique for every state $s \in S$. □

Next we define another collection of sets of "states" that represents an aggregation of the state space. Define S_r^0 to be the aggregated states of rank r to be $\{\mathbf{x} = (x_1, \ldots, x_r) \in \mathbb{Z}^r \mid D \geqslant x_1 \geqslant x_2 \geqslant \cdots \geqslant x_r \geqslant r\}$. Thus the elements of S_r^0 are simply the elements of S_r shortened to be vectors of length r. Of course duplicates are removed. Recall that $S_0 = \{\mathbf{0}\}$, and we need an aggregated state corresponding to state 0, so define S_0^0 to be the set containing this one aggregated state which can be thought of as an *empty* vector, one with no components. Now define $S^0 = \bigcup_{r=0}^C S_r^0$. The set S^0 is the state space of aggregated states. We will use the letters $\mathbf{x}$ and $\mathbf{y}$ to denote aggregated states and the letters $\mathbf{s}$ and $\mathbf{t}$ to denote normal states. Finally, define $d(\mathbf{x})$ as the dimension of vector $\mathbf{x}$ so that $d(\mathbf{x}) = r$ for all $\mathbf{x} \in S_r^0$, i.e., the size of the vector $\mathbf{x}$. We now can calculate how much smaller the aggregated state space S^0 is than the full state space S.

PROPOSITION 3. *Let D be the maximum LOS, C the capacity of the ICU, and M be the maximum number of arrivals on one day. Then*

$$|S| = \binom{D+C}{C}, \quad |A| = \binom{D+M}{M}, \quad and$$

$$|S^0| = \sum_{r=0}^{\min(D,C)} \binom{D}{r} \leqslant 2^D.$$

PROOF. See Appendix A of the online companion, which is available as part of the online version that can be found at http://or.journal.informs.org/. □

For example with $D = 6$, $C = 15$, we have $|S| = 54{,}264$ and $|S^0| = 64$.

We now define sets of intermediate states that will allow us to compute the stationary distribution for the states S from the stationary distribution for states S^0. Define S_r^l to be the aggregated states of rank $r \in \{0, \ldots, C\}$ and level $l \in \{0, \ldots, C - r\}$ to be

$$S_r^l = \big\{ \mathbf{x} = (x_1, \ldots, x_r, \ldots, x_{r+l}) \in \mathbb{Z}^{r+l} \mid D \geqslant x_1 \geqslant \cdots \geqslant x_r$$
$$\geqslant r \geqslant x_{r+1} \geqslant \cdots \geqslant x_{r+l} \geqslant 0 \big\}.$$

The set S_r^l is just the disaggregated states of S_r shortened to length $r + l$, with of course duplicates removed. For example, the state $(11, 10, 6, 5, 2, 1)$ is a state with rank 4 and level 2. For notational simplicity we define $S_r^l = S_r$ for $l = C - r, \ldots, C$, to be different names for the same set, namely S_r. Next define $S^l = \bigcup_{r=0}^C S_r^l$. As before, one can demonstrate that $\{S_r^l\}_{r=0}^C$ is a partition of S^l.

Later, it will be useful to look at subsets of the states S or S_r or the aggregated states S^l or S_r^l, where the first components of the state matches another vector. Define $S(\mathbf{v}) = \{\mathbf{s} \in S \mid s_i = v_i, i = 1, \ldots, n\}$ for vector $\mathbf{v} \in \mathbb{Z}^n$. Define $S_r(\mathbf{v})$, $S^l(\mathbf{v})$, and $S_r^l(\mathbf{v})$ analogously.

We now create a Markov chain on the set of aggregated states S^l and define the transition matrix $\mathbf{P}^l$ by

$$P_{xy}^l = \sum_{u=0}^{o(x)} P_o(u, o(\mathbf{x})) \sum_{a \in A_{(x, u)y}^l} p(\mathbf{a}),$$

where $A_{(x, u)y}^l$ is defined as the set of arrival vectors $\mathbf{a} \in A$ that makes the chain transition from aggregated state $\mathbf{x} \in S^l$ to aggregated state $\mathbf{y} \in S^l$ conditioned on preserving u outliers, namely

$$A_{(x, u)y}^l = \{\mathbf{a} \in A \mid f_{d(y)}(\mathbf{a}, (\mathbf{x}, u)) = \mathbf{y}\}, \quad \text{for } \mathbf{x}, \mathbf{y} \in S^l. \quad (2)$$

Analogously, we can define π^l be the solution to $\pi^l = \pi^l \mathbf{P}^l$ and $\sum_{x \in S^l} \pi_x^l = 1$.

The aggregation procedure has allowed us to reduce the size of the problem so that it is computationally feasible to compute a stationary probability distribution, π^0, for the aggregated states of level 0. This distribution is, of course,

useless, unless it will allow us to find the stationary probability π for the (disaggregated) states. Note that we could have aggregated the states in many ways and calculated an associated stationary distribution. The value of the particular aggregation (to level 0) is that it can be disaggregated exactly to compute π with less computation and less memory usage than computing π directly. The disaggregation process proceeds by levels.

Given π^0, we now compute $\pi^1, \ldots, \pi^{C-1} \equiv \pi$ in turn by an algorithm described below, which computes π^l from π^i for $i = 0, \ldots, l - 1$. By definition, π^l satisfies

$$\pi^l = \pi^l \mathbf{P}^l,$$

or for $\mathbf{y} \in S^l$ we know that

$$\pi_y^l = \sum_{x \in S^l} \pi_x^l P_{xy}^l = \sum_{x \in S^l} \pi_x^l \sum_{u=0}^{o(x)} P_o(u, o(x)) \sum_{a \in A_{(x, u)y}} p(\mathbf{a}).$$

First we interchange the sums to obtain

$$\pi_y^l = \sum_{a \in A} p(\mathbf{a}) \sum_{u=0}^{D} \sum_{\{x \in S^l, f(a,(x,u))=y\}} \pi_x^l P_o(u, o(x)). \quad (3)$$

To calculate (3) we need to order the computation so that the relevant π_x^l needed for calculating π_y^l have already been determined. We accomplish this by using the following ordering: π^l is computed before π^{l+1} for $l = 0, \ldots, C - 2$. Within one level we need to compute π_x^l, and compute the stationary probability for states $\mathbf{x} \in S_r^l$, i.e., with rank r, before those of rank $r + 1$ for $r = 0, \ldots, D$. Within one rank, i.e., those in S_r^l, we order them in decreasing lexicographic order, i.e., $\mathbf{x}$ is computed before $\mathbf{y}$ if $\mathbf{x}$ is lexicographically greater than $\mathbf{y}$, namely if the first nonzero component of $\mathbf{x} - \mathbf{y}$ is positive. We denote this as $\mathbf{x} \overset{L}{\succ} \mathbf{y}$.

To preserve the ordering, we must assume that at most D outliers can stay as outliers in the beginning of the next period. Thus we modify the probability $P_o(u, o(\mathbf{x}))$ to $P_o(u, \min\{o(\mathbf{x}), D\})$. The assumption is not that restrictive because if the probability of more than D outliers were significant, the ICU would be severely overloaded. It is instructive to consider some specific examples. Suppose we know π^0 and π^1. Thus we have performed the disaggregation to level 1 and are in the middle of the computation of π^2. Let $(D = 10)$, and as a first example let the (destination) state be

$$\mathbf{y} = (10, 10, 9, 5, 2, 1),$$

which is a state of rank 4 and level $l = 2$ because $y_4 \geqslant 4 \geqslant y_5$. Now pick an arrival vector $\mathbf{a} = (10, 10)$. This arrival will make up some of the components of the destination state, in this case two, and the other components of the destination state must come from the departure state $\mathbf{x}$. The departure state must be of the form $\mathbf{x} = (10, 6, 3, 2, \ldots)$.

Whatever the additional components of **x** are, past component 4, they will not affect the destination state (at this level of disaggregation), so we will consider only 4-vectors and call **v** $= (10, 6, 3, 2)$. One way to compute the appropriate sum of π_x^l's would be to look at $S^l(\mathbf{v}) = \{\mathbf{x} \in S^l \mid x_i = v_i, i = 1, \ldots, d(\mathbf{v})\}$. For the examples that follow we show that the set $S^l(\mathbf{v})$ can be aggregated in some way so that the sum of π_x^l's only requires ones that have been calculated already, i.e., π_x^m for $m < l$ or if $m = l$, then $r(\mathbf{x}) < r(\mathbf{y})$, or if the ranks are equal, then $\mathbf{x} \overset{L}{\succ} \mathbf{y}$. For the case that **v** is an aggregated state, the crucial question is what is its rank and level. For this example, **v** is clearly rank 3 level 1, because $v_3 \geqslant 3 \geqslant v_4$. Thus for the state **y** and the arrival **a**, $S^2(\mathbf{v}) = \{(10, 6, 3, 2, u_5) \mid u_5 = 0, 1, 2\}$ is a collection of rank 3, level 2 states. Now observe that we do not need the values of $\{\pi_x^2\}_{x \in S^2(\mathbf{v})}$. We can simply use $\pi_\mathbf{v}^1$ which is already computed because $\mathbf{v} = (10, 6, 3, 2)$ is a rank 3, level 1 state and we computed level 1 before level 2.

As a second example, if **y**, the destination state, is $(10, 10, 9, 5, 1, 1)$ and $\mathbf{a} = (10, 10)$, then $\mathbf{v} = (10, 6, 2, 2)$ and we would access $\pi_\mathbf{v}^2$. This state **v** is a rank 2 level 2 state and we compute rank 2 before rank 4.

As a third example, if $\mathbf{y} = (10, 10, 9, 5, 2, 1)$ and there are no arrivals, then the departure state is $\mathbf{x} = (10, 10, 10, 6, 3, 2)$, where two among three outliers remain as outliers. The state **x** is rank 4 level 2, which is the same rank and level of **y**, yet $\mathbf{x} \overset{L}{\succ} \mathbf{y}$, so we computed π_x^l before π_y^l.

As a fourth example, if $\mathbf{y} = (10, 10, 9, 5, 2, 1)$ and there are five arrivals $\mathbf{a} = (10, 10, 9, 2, 1)$, then the departure state must generate $y_4 = 5$. Thus $S^2((6))$ is a set of vectors of the form $(6, \ldots)$. In this case, we need to access π^0 (all previously calculated) for a set of aggregate (level 0) states of different ranks:

Rank 1 state (6) and

Rank 2 state (6, 2).

As a final example, if the destination state is $\mathbf{y} = (10, 10, 10, 1)$ and $\mathbf{a} = (1)$, then the departure state is either $\mathbf{x}_0 = (10, 10, 10, 0)$, $\mathbf{x}_1 = (10, 10, 10, 1)$, or $\mathbf{x}_2 = (10, 10, 10, 2)$, where all three outliers are preserved. Following our ordering, $\mathbf{x}_2 \overset{L}{\succ} \mathbf{y}$ has been computed already but both $\mathbf{x}_0 \overset{L}{\prec} \mathbf{y}$ and $\mathbf{x}_1 \overset{L}{\prec} \mathbf{y}$ so we do not know $\pi_{\mathbf{x}_0}^1$ or $\pi_{\mathbf{x}_1}^1$. However we can still calculate them from $\pi_{(10,10,10)}^0 - \pi_{(10,10,10,3)}^0$, in which we calculated the probability for $(10, 10, 10, 3)$ before this step. This particular case can happen only in the level 1 disaggregation, in which **v** takes the following form: $\mathbf{v} = (D, D, \ldots, D, a)$, where $a < r(\mathbf{v})$. See the proof of Proposition 4 in the online appendix for more details.

What we need to ensure does not happen is that if, for example, $\mathbf{y} = (10, 10, 9, 5, 2, 1)$, a rank 4 level 2 state, two arrivals occur, and the associated vector **v** is a 4-vector, then one of the departure states **x** cannot be a rank 1 level 3 state because we have not computed any stationary probabilities for level 3, yet and if it is rank 5 or 6, then its level must be less (1 or 0, respectively).

Stated more generally, if a destination state is of rank r level l and the arrival consists of k arrivals then the departure vector, **x**, is a $(r + l - k)$-vector and we must ensure that it is associated with a state or set of states of level less than l, or level equal to l and rank less than r, or level equal to l and rank equal to r but lexicographically larger.

Define $\mathbf{v}(\mathbf{a}, \mathbf{y})$ as follows. Let **y** be a destination state and **a** an arrival vector. First remove the components of **a** from the components of **y**. Note that some components of **a** may not appear in **y** because they are smaller than the smallest component of **y**. The remaining vector will have $u = o(\mathbf{y}) - o(\mathbf{a})$ components equal to D, which will be the first components of $\mathbf{v}(\mathbf{a}, \mathbf{y})$. The remaining components less than D will be incremented by 1 and added to $\mathbf{v}(\mathbf{a}, \mathbf{y})$.

PROPOSITION 4. *If we compute $\pi^1, \pi^2, \ldots, \pi^C = \pi$ in that order and for each vector, we compute $\pi_\mathbf{x}^l$ for $\mathbf{x} \in S^l$ based on the order given to states—in increasing order of rank and then decreasing lexicographic order of vectors within rank—then it is possible to compute*

$$\pi_y^l = \sum_{a \in A} p(a) \sum_{x \in S^l(\mathbf{v}(\mathbf{a},\mathbf{y}))} \pi_x^l P_o\big(o(\mathbf{y}) - o(\mathbf{a}), \min\{o(\mathbf{x}), D\}\big)$$

in a way that only uses quantities computed before π_y^l.

PROOF. See the online Appendix A. □

5. Comparison of Computational Methods

Calculating the stationary probability distribution requires substantial computer time and storage, no matter what method is employed. We demonstrate here why our specialized aggregation-disaggregation method requires substantially less computer storage and less computer time than the standard Gauss-Seidel iterative method for solving $\pi = \pi\mathbf{P}$ and $\sum_{s \in S} \pi_s = 1$. Furthermore, the advantage in the storage requirement improves as the size of the problem grows.

Computational Complexity

For the Gauss-Seidel iterative method, the time requirements are $O(K|S|^2)$ (Bolch et al. 2006), where $|S|$ is the number of ordered states equal to $\binom{D+C}{C}$ and K is the number of iterations required for convergence. For our problem, the time required for the iterative procedure to converge is dominated by the time to create the transition matrix, which is $O(C|A||S|^2)$, where we recall that $|A| = \binom{D+M}{M}$ is the number of arrival patterns. This amount of computation is required because, for each departure state and for each arrival pattern, we must sort the combined vector of size C and add the associated probability to the appropriate destination state of the transition matrix.

To derive formulas for the time and space requirements of the aggregation-disaggregation procedure strictly in terms of $|S|$ leads to rather loose bounds. Thus, we derive more precise (although more complex) formulas in terms

of other problem parameters: D, M, and C, and then show numerically the comparison to Gauss-Seidel for sample values of these parameters. First, we need to know the number of $(\mathbf{x}, u)$ combinations for $\mathbf{x} \in S^0$.

LEMMA 1. *Let* $U = \{(\mathbf{x}, u) \mid \mathbf{x} \in S^0, \; u \leqslant o(\mathbf{x})\}$ *then* $|U| < 2^{D+1}$.

PROOF. First, we need to calculate the number of aggregate states with exactly i outliers. We claim that this is bounded by 2^{D-1-i}, which is the number of aggregated states if the maximum LOS was $D - 1 - i$. If $i = 0$, then the states can have a maximum value of $D - 1$ and so by Proposition 3 the number of states is bounded by 2^{D-1}. If there are $i > 0$ outliers, then we need to look at all possible patterns after the outliers in the remaining components of the vector. We further restrict our attention, temporarily, to states of rank r, and so there are exactly $r - i$ positions which can each take on values $D - 1, \ldots, r$. Applying the same argument from Proposition 3 we can subtract $j = 0, \ldots, r - i - 1$, respectively, from each of the remaining $r - i$ positions and the collection of such objects is exactly like choosing $r - i$ objects from a set of size $D - 1 - i$. Summing over the ranks $r = i, \ldots, D - 1$ we obtain

$$\sum_{r=i}^{D-1} \binom{D-1-i}{r-i} = \sum_{j=0}^{D-1-i} \binom{D-1-i}{j} = 2^{D-i-1}.$$

There is only one state with D outliers. Now for each of the states the possible values for u are $0, \ldots, i$ for a state with i outliers. Thus,

$$|U| = \sum_{i=0}^{D-1}(i+1)2^{D-1-i} + (D+1) = \sum_{j=1}^{D} j 2^{D-j} + (D+1)$$

$$= \sum_{j=1}^{D}\sum_{k=1}^{j} 2^{D-j} + (D+1) = \sum_{k=1}^{D}\sum_{j=k}^{D} 2^{D-j} + (D+1)$$

$$= \sum_{k=1}^{D}(2^{D-k+1} - 1) + (D+1) = 2\sum_{k=1}^{D}(2^{D-k}) + 1$$

$$\leqslant 2^{D+1} - 1. \quad \square$$

The aggregation step, to compute $\mathbf{P}^0$, requires $O(|2^D||2^{D+1}|\binom{D+M}{M})$ time since there are 2^D aggregated states (at level 0), and for each pair of states $((\mathbf{x}, u), \mathbf{y})$ we need to compute $P^0_{(x,u)y}$ which requires examining $\binom{D+M}{M}$ arrival patterns (see the online Appendix A). To solve the equations $\pi^0 = \pi^0 \mathbf{P}^0$, $\sum_{x \in S^0} \pi_x^0 = 1$, for π^0 requires an $O(|2^D|^3)$ calculation to apply Gaussian elimination to the aggregated system.

Computing the time for the disaggregation procedure is more complex. For each level, $l = 1, \ldots, C - 1$, we consider all the states $\mathbf{x}$ at level l and find π_x^l. Some of these are easy. For instance, if state $\mathbf{x}$ ends with two 0's, then $\pi_x^l = \pi_{\bar{x}}^{l-1}$ where $\bar{\mathbf{x}} \in S^{l-1}$ is the state associated with $\mathbf{x}$ with only one 0 at the end. So to determine the number of states for which we need to do additional work, we first need the following lemma.

LEMMA 2. *Define* $g(r, l) = \binom{\min(r+l, C)}{r}$. *If* $\mathbf{x} \in S_r^0$ *then* $|S^l(\mathbf{x})| = g(r, l)$.

PROOF. Let $\mathbf{y} \in S^l(\mathbf{x})$. Let $(z_1, \ldots, z_n)$ be the components of y past position r: $z_i = y_{i+r}$. So $n = \min(r + l, C) - r$. Transform z into z' by $z_i' = z_i + n - (i - 1)$. Because the z_i's were in decreasing order the z_i''s are distinct and $1 \leqslant z_i' \leqslant r + n$, so the number of possible distinct z' vectors is $\binom{n+r}{n} = \binom{n+r}{r} = g(r, l)$. $\quad \square$

From Lemma 2 the number of states in S^l associated with $\mathbf{x} \in S_r^0$ with a 0 at the end is $g(r, l - 2)$. Thus the total number of states we have to do more than $O(1)$ work for is

$$\sum_r \sum_l \binom{D}{r}(g(r, l) - g(r, l-2)),$$

where r ranges over possible ranks, l ranges over possible levels for that rank, and recall that $\binom{D}{r} = |S_r^0|$. For these states we need to consider all possible arrival patterns of which there are $\binom{D+M}{M}$. Given a destination state and arrival, we can identify the origination state using a calculation that is linear in C. Accessing the origination state's probability is $O(1)$ unless you need some combination of probabilities from S^0. In that event the worst case is a $O(2^{D+1})$ calculation. Thus the running time of the disaggregation procedure is bounded by

$$O\left(\sum_r \sum_l \binom{D}{r}(g(r, l) - g(r, l-2))\binom{D+M}{M}\right.$$

$$\left. \cdot C \max\left(2^{D+1}, \max_{r'} r' \cdot g(r', l)\right)\right).$$

Because $\max(2^{D+1}, \max_{r'} r' \cdot g(r', l)) \leqslant |S|$, the bound can be relaxed to

$$O\left(C|S|\binom{D+M}{M}\sum_r \binom{D}{r}\sum_l(g(r, l) - g(r, l-2))\right).$$

By summing over l, we also can relax $\sum_l(g(r, l) - g(r, l-2)) \leqslant g(r, C) + g(r, C-1) \leqslant 2g(r, C)$. We also know that $\sum_r \binom{D}{r} g(r, C) = |S|$. Therefore, the bound of the running time of the disaggregation procedure becomes

$$O\left(C|S|^2\binom{D+M}{M}\right),$$

and this bound is the same as that on the setup of the matrix for the Gauss-Seidel algorithm. Therefore, we see that theoretically the entire running time for our algorithm is less than the setup time for Gauss-Seidel. We also note that our algorithm is an exact procedure while Gauss-Seidel is iterative and must converge.

Storage Requirements

For Gauss-Seidel the storage requirements are $O(|S|^2)$. The aggregation-disaggregation procedure requires $O(|S|)$ to store the vector of stationary probabilities and $O(2^{D+1}\binom{D+M}{M})$ to store a probability for each state in S^0 (including u) and each arrival vector. Thus, the storage requirement is $O(\max(|S|, 2^{D+1}\binom{D+M}{M}))$. In the following we show, with numerical examples, that the difference in storage requirements is very large and renders Gauss-Seidel infeasible for realistically sized problems.

Actual Computational Comparison

To show that our previous theoretical analysis of relative computational complexity holds in practice, we compare our procedure (AD) with Gauss-Seidel (GS) for actual problems. Note that we are considering seven-day schedules and that for both procedures we remove the many states that cannot be reached. For a ten-bed example running on a PC with two quad-core processers at 3.16 GHz each and 16 GB of RAM, AD required 9 minutes when $M = 7$; the procedure to set up the matrix for GS took over 15 minutes with a total running time to calculate the stationary probability distribution of about 16 minutes; that is, GS took about twice as long.

The storage requirements were 0.04 GB for our algorithm and 0.51 GB for GS, i.e., more than 10 times as much storage. For problems with more beds we can only estimate the Gauss-Seidel storage requirements because they are too large to actually run the algorithm. We estimate the storage requirements for GS by determining the number of states in the Markov chain and then aggregating those that transition to the same states. For example, in a five-bed ICU, the states $(3, 3, 3, 1, 1)$, $(3, 3, 3, 1, 0)$, and $(3, 3, 3, 0, 0)$ transition to $(2, 2, 2, 0, 0)$ at the beginning of the next day. Thus we do not need the stationary probabilities of the three states, but the sum of the stationary probabilities of the three. In the case of the ten-bed ICU this leads to a state space reduction from $|S| = 8{,}008$ to $3{,}003$ for one day. To calculate bumping rates for a seven-day schedule using double precision (8 bytes per number), the GS algorithm will need $|S^2|$ $(7)(8$ bytes$)$ or 0.5 GB. This is very close to what we observed when running the algorithm. Therefore, we are confident that our estimates for memory usage in the bigger problems are accurate.

In Table 2 we compare the RAM requirements of the two methods for different problem sizes. We vary the number of beds from 10 to 16 and vary the maximum number of new arrivals per day from 5 to 8. If the average LOS

Table 2. Physical memory requirements (gigabytes) of the aggregation-disaggregation procedure and the Gauss-Seidel procedure.

RAM requirements	AD algorithm				Gauss-Seidel
	Maximum number of arrivals				
	$M = 5$	$M = 6$	$M = 7$	$M = 8$	
Number of beds					
$C = 10$	0.030	0.032	0.035	0.040	0.51
$C = 11$	0.031	0.033	0.036	0.041	1.07
$C = 12$	0.032	0.034	0.037	0.042	2.14*
$C = 13$	0.033	0.035	0.039	0.043	4.11*
$C = 14$	0.034	0.036	0.040	0.044	7.57*
$C = 15$	0.035	0.037	0.041	0.045	13.46*
$C = 16$	0.036	0.038	0.042	0.046	23.19*

*Estimates.

Table 3. Running time (in hours) of the aggregation-disaggregation procedure and the Gauss-Seidel procedure.

Running time (hours)	Maximum number of arrivals							
	$M = 5$		$M = 6$		$M = 7$		$M = 8$	
	AD	GS	AD	GS	AD	GS	AD	GS
Number of beds								
$C = 10$	0.03	0.10	0.08	0.17	0.15	0.27	0.30	0.50
$C = 11$	0.10	0.25	0.22	0.42	0.43	0.70	0.80	1.15
$C = 12$	0.25	—	0.53	—	1.15	—	2.00	—
$C = 13$	0.57	—	1.25	—	2.5	—	4.5	—
$C = 14$	1.25	—	2.83	—	5.5	—	10.33	—
$C = 15$	2.58	—	5.67	—	11.47	—	21.5	—
$C = 16$	5.08	—	11	—	24.5	—	42.5	—

—: Not available.

is approximately 4 days, a 16-bed ICU operating at 90% utilization would average 3.6 arrivals per day. Setting the maximum daily arrivals at more than twice this average gives us the ability to capture demand surges. For 10 and 11 beds we could run both algorithms and report the actual memory usage by the computer. We could not run GS for more than 11 beds, and thus we only estimate the memory usage based on our earlier theoretical analysis. For our AD algorithm we could run all the cases and report the actual memory usage. We can see that as the problem size increases, the AD algorithm memory usage remains low and very stable. However, GS requires much more memory, and this need rapidly increases with problem size.

Table 3 displays similar results for running time. We see that the AD algorithm is significantly faster than GS and can be used for realistic problem sizes that GS cannot handle because of memory usage. We also see that, from the perspective of running time, with a standard desktop computer, an approximate upper limit of 15 ICU beds is practical to analyze using our algorithm if one is trying many different scenarios. Iapichino et al. (2004) show a mean number of ICU beds of 9.6 for 89 adult ICUs in European countries, and Kirchhoff and Dahl (2006) show a mean of 15 beds with 38% of adult ICUs having less than 12 beds (118 units in the United States). Groeger et al. (1992) found the mean ICU size to be 11.7 beds in the United States. Given this data on ICU sizes, we believe that our algorithm is currently applicable to the majority of ICUs.

6. Performance Measures

The purpose of the Markov chain model we have formulated is to evaluate how well an ICU performs for different patient arrival patterns and capacity/load scenarios. We consider three measures of performance:

P_B = probability that a randomly arriving patient is bumped sometime in their stay;

R = the expected number of days remaining for a patient that is bumped;

ρ = the utilization of the ICU capacity.

To compute P_B, define the function q_{as} to be the number of patients bumped with an arrival **a** in state **s**. Thus, given state **s** and an arrival vector **a** with m arrivals, define **v** to be $f_{C+m}(\mathbf{a}, (\mathbf{s}, u))$. The last m positions of the vector **v** correspond to the patients being bumped if they are positive. Thus

$$q_{as} = \sum_{k=1}^{m} \mathbf{1}_{\{v_{C+k} > 0\}}.$$

Given this function, we can compute the probability of a bump occurring, P_B, by conditioning on the arrivals and states and dividing by the expected number of arrivals in a day:

$$P_B = \sum_{a \in A} \sum_{s \in S} q_{as} p(\mathbf{a}) \pi_s / E(N).$$

To compute the expected days remaining in a bumped patient's stay, we define R_{as} to be the number of days bumped when we get an arrival **a** in state **s**. We also define $P(\mathbf{a}, \mathbf{s} \mid Bump)$ to be the probability of being in state **s** and having arrival vector **a** in state **s** given that this pair generates a bump, and we define B to be the set of pairs **a**, **s** that generate at least one bump. Given these definitions, our performance metric is

$$R = \sum_{(a, s) \in B} \frac{R_{as}}{q_{as}} P(\mathbf{a}, \mathbf{s} \mid Bump).$$

The utilization is straightforward, namely,

$$\rho = \frac{1}{C} \sum_{s \in S} \pi_s \left(\sum_{i=1}^{C} \mathbf{1}_{\{s_i > 0\}} \right).$$

7. Computational Experiments

Scenarios

Our main intent for this computational experiment was to understand the implication of a more seasonal arrival

Table 4. The mean number of arrivals for all the arrival patterns.

		Surgery day	Nonsurgery day
A	3-day	5.79	0.90
	5-day	3.84	0.90
	7-day	3.00	
B	3-day	5.00	1.50
	5-day	3.60	1.50
	7-day	3.00	
C	3-day	4.21	2.10
	5-day	3.36	2.10
	7-day	3.00	

pattern driven, in part, by a surgical block schedule versus a more uniform schedule. Thus we considered three alternatives: a three-day surgical schedule in which all arrivals to the ICU from scheduled surgeries would occur Monday, Wednesday, or Friday; a five-day surgical schedule in which all arrivals to this ICU from scheduled surgery would occur Monday through Friday; and finally a seven-day surgical schedule in which arrivals would occur in a stochastically stationary manner each day. The arrival distributions for scheduled and unscheduled patients were constructed so that the mean daily arrival across all days was three patients and the maximum number of arrivals is seven. The detailed arrival probability mass functions appear in the online appendix. To understand how robust these results were, we adjusted a number of factors, including the percentages of scheduled and unscheduled arrivals, the LOS distribution (see Figure 1), and the overall load factor on the ICU. For the percentage of scheduled and unscheduled arrivals we considered, respectively, the following cases: (A) 70%/30%, (B) 50%/50%, and (C) 30%/70%.

The mean daily arrivals for each arrival pattern are shown in Table 4. The third and fourth columns display the mean arrivals for the surgery day and nonsurgery day. Note that the unscheduled arrival distributions for three-day, five-day, and seven-day are the same. By comparing the surgery and nonsurgery days, we get a sense for how strong the seasonality is in each scenario. We see that three-day schedules create the greatest peaks in arrivals. Scenario C, with

Figure 1. Length of stay distribution.
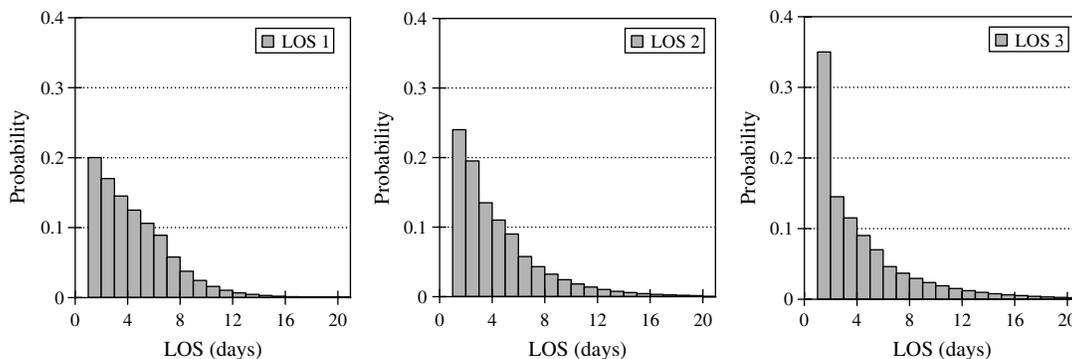
the highest proportion of unscheduled arrivals, leads to the smoothest pattern across days.

We consider a mean LOS of four days with three LOS distributions, where the coefficients of variation are 0.70 ($P^O = 0.65$), 0.86 ($P^O = 0.75$), and 1.03 ($P^O = 0.80$), respectively. The LOS distributions are depicted in Figure 1. Weissman (1997) identified that the mean LOS is skewed by the outliers (patients with extremely long LOS). 88.7% of patients have a LOS less than seven days. (Table 2 in Tu and Mazer 1996 and Figure 1 in McManus et al. 2004 also show the similar phenomenon.) Thus in our LOS distributions, we consider six days as the maximum base LOS with longer LOS generated by a geometric LOS process for the outliers (those arriving with LOS = 6). There are many studies of ICU LOS giving a range of results. Knaus et al. (1993) in a study of 17,440 ICU admissions find a mean LOS of 4.7 across different regions of the United States. Marik and Hedman (2000) show a mean LOS of 2.8 days for the 750 ICU admissions during a six-month period and also show a mean LOS of 2.3 days for the surgical ICU and 3.1 days for the medical ICU. Given this data, we use a mean LOS of four days as representative while acknowledging that LOS distributions will vary across ICU types and hospitals.

The empirical literature report a wide range of coefficients of variation for ICU LOS. For example, Weissman (1997) reports values ranging from 0.4 to 2.2, depending on the type of surgery. The three example LOS distributions we use fall in this range.

To adjust the load factor, we adjusted the number of beds and considered ICUs with 13, 14, and 15 beds corresponding to load factor of 0.92, 0.86, and 0.80, respectively. As discussed previously, the number of beds we considered is representative of many ICUs.

The results for these 81 cases are displayed in the three graphs in Figure 2. In each case the probability of being bumped is plotted. The cases encoded on the category axis are labeled $X - b$ where $X = A, B, C$ is the various percentages for scheduled/unscheduled patients, and b is the number of beds equal to 13, 14, 15. The results for the three-day, five-day, and seven-day scenarios are plotted separately.

The first phenomenon to notice is that, as expected, increasing capacity significantly reduces percentage of patients who are bumped, but with diminishing returns. Second, somewhat surprisingly, three-day schedules lead to less bumping than the less seasonal schedules. We believe the reason for this is that the gaps between surgery days gives the system more time to clear out patients. The surgery days with three-day schedules have the most bumping (see Figure 3), but those days are fewer than in the five-day and seven-day schedules and are spread apart. This means that you are unlikely to get two days in a row of many arrivals. Third, we see that moving from scenarios A to C, i.e., increasing the proportion of unscheduled arrivals, tends to increase the bumping rate. This phenomenon has

Figure 2. (I): Average of the probability of being bumped for LOS distribution type 1; (II): Average of the probability of being bumped for LOS distribution type 2; (III): Average of the probability of being bumped for LOS distribution type 3.
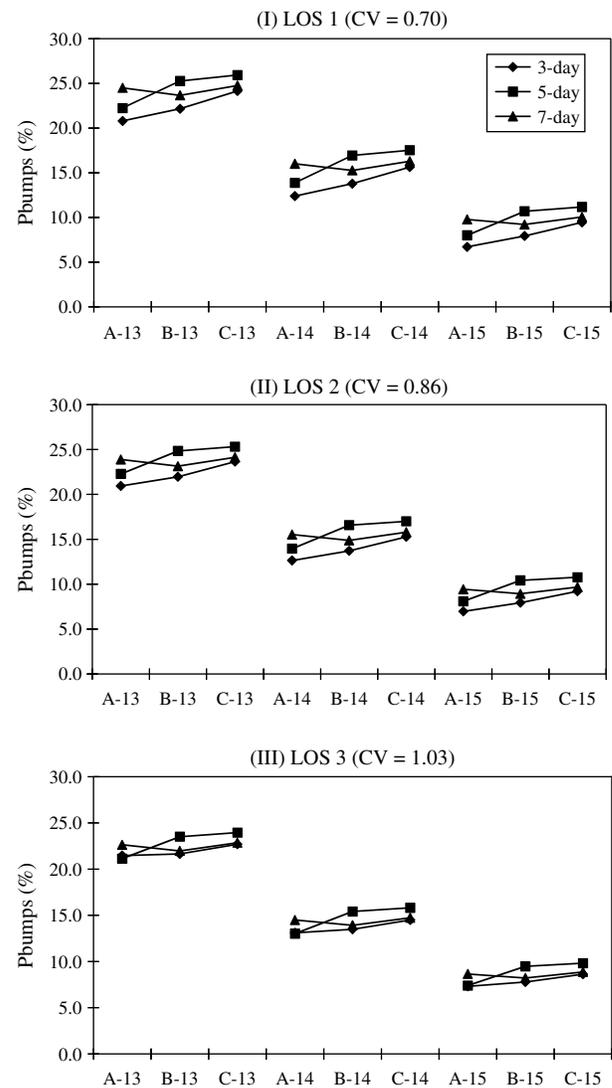
a similar explanation to the previous one. Increasing the proportion of unscheduled arrivals has two effects. On one hand, there is less seasonality: the days with the most bumps are less extreme than other days, as can be seen in Figure 3. However, at the same time there is no time for the system to recover from a large number of arrivals because any day could potentially have many arrivals. Thus on average more unscheduled arrivals lead to more bumping, as can be seen in Figure 2. Finally, we see that the differences in the LOS distributions we use have only a minor effect.

As the graphs show, our algorithm could be used to evaluate options for reducing bumping and explore some interesting trade-offs faced by hospital administrators. For example, looking at the A-13 case, we see that moving

Figure 3. The probability of being bumped on the worst day across the week for LOS distribution type 1.

LOS 1 (CV = 0.70)
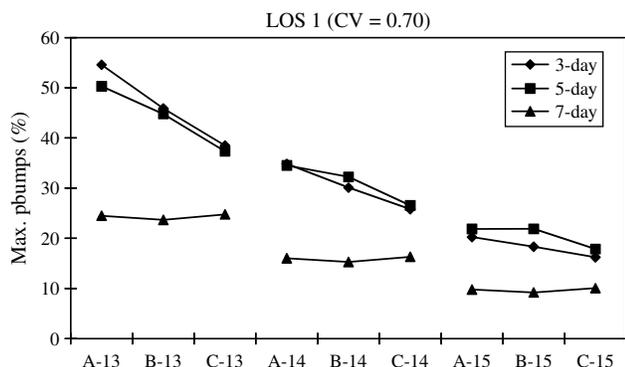
from a three-day schedule to a seven-day schedule will increase the bumping rate by approximately 4% (in absolute terms) while decreasing it by 30% on the worst days. A hospital may prefer the seven-day schedule because large numbers of simultaneous bumps might be very disruptive. In comparing these results to the A-14 case, we see that adding a bed reduces the average bumping by about 9% (in absolute terms) and about 20% on the worst days. These results suggest that smoothing the surgery schedule can have a more significant impact on bumping rates on the worst days than increasing capacity. Thus, by using our model, the hospital administration could evaluate whether it makes more sense to try to even out the surgery schedule or pay for the added beds (and staff).

The other measure we collect is the expected remaining length of stay if one were bumped. The expected LOS remaining on the day the patient was bumped was surprisingly uniform varying across all scenarios—always close to 1.1 days. The same was true for the expected LOS on the days with the most bumps. These results make sense because a large percentage of patients have a LOS of two days or fewer, and thus the policy of bumping the healthiest patients leads to bumps being mainly of patients with one day remaining.

8. Conclusion

We have modeled an ICU as a Markov chain in which the state indicates the remaining length of stay of each patient, and we can thus more accurately study the effect of ICU workload on patient bumping. Although the usual method of ordering the components reduces the state space dramatically, the state space is still too large to use to compute the stationary distribution, π, using the standard Gauss-Seidel technique for reasonable ICU parameters. We employ an aggregation technique that reduces the size of the state space so that computation of the stationary distribution of this aggregated Markov chain is possible. In addition, this particular aggregation allows us to disaggregate the stationary distribution exactly, albeit with substantial computation.

Yet, for moderately sized ICUs the computational time and space are substantially less than the alternative standard method, Gauss-Seidel.

We then investigate how surgical schedules (for elective procedures), a factor under the control of the hospital, can affect the performance of the ICU. We investigate how three-day, five-day, and seven-day surgical schedules work across a variety of scenarios that differ in load, in the coefficient of variation of LOS, and the percentage of scheduled cases. Although the qualitative results along any factor (e.g., lowering utilization) are predictable, the tool does provide administrations with the capability to understand the quantitative trade-offs between, for example, adding one more bed versus shifting the schedule of surgeries.

Several avenues suggest themselves for future work. First, it may be possible to improve on the computational time required to disaggregate π^0 to π exactly. Second, we have modeled a single ICU, but major hospitals have many ICUs, which interact in periods of high utilization by passing patients from one to another, making room for new arrivals. Being able to model multiple ICUs simultaneously would allow hospital management to investigate various issues such as, should two surgical ICUs, say, a general surgery ICU and a cardiac-thoracic unit be combined or remain separate? If they remain separate, is there any value to cross training some nursing staff? By cross training nursing staff across two or more ICUs, one could effectively combine the ICUs, and bumping from one to another would not have the same effect on the quality of care.

9. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

References

Akcali, E., M. J. Côtè, C. Lin. 2006. A network flow approach to optimizing hospital bed capacity decisions. *Health Care Management Sci.* **9**(4) 391–404.

Bellandi, D., C. Rauber. 1999. Running at capacity. *Modern Healthcare* **29**(25) 110–113.

Bolch, G., S. Greiner, H. de Meer, K. S. Trivedi. 2006. *Queueing Networks and Markov Chains*, 2nd ed. John Wiley & Sons, Hoboken, NJ.

De Bruin, A. M., A. C. van Rossum, M. C. Visser, G. M. Koole. 2007. Modeling the emergency cardiac in-patient flow: An application of queuing theory. *Health Care Management Sci.* **10**(2) 125–137.

De Véricourt, F., O. B. Jennings. 2008. Nurse-to-patient ratios in hospital staffing: A queuing perspective. ESMT research working papers, ESMT-08-005, https://www.esmt.org/fm/479/ESMT-08-005.pdf.

Friedman, B., C. Steiner. 1999. Does managed care affect the supply and use of ICU services? *Inquiry* **36**(1) 68–77.

Green, L. V. 2002. How many hospital beds? *Inquiry* **39**(4) 400–412.

Green, L. V., V. Nguyen. 2001. Strategies for cutting hospital beds: The impact on patient service. *Health Service Res.* **36**(2) 421–442.

Groeger, J. S., M. A. Strosberg, N. A. Halpern, R. C. Raphaely, W. E. Kaye, K. K. Guntupalli, D. L. Bertram, et al. 1992. Descriptive analysis of critical care units in the United States. *Critical Care Medicine* **20**(6) 846–863.

Gruenberg, D. A., W. Shelton, S. L. Rose, A. E. Rutter, S. Socaris, G. McGee. 2006. Factors influencing length of stay in the intensive care unit. *Amer. J. Critical Care* **15**(5) 502–509.

Harrison, G. W., A. Shafer, M. Mackay. 2005. Modelling variability in hospital bed occupancy. *Health Care Management Sci.* **8**(4) 323–334.

Iapichino, G., L. Gattinoni, D. Radrizzani, B. Simini, G. Bertolini, L. Ferla, G. Mistraletti, F. Porta, D. R. Miranda. 2004. Volume of activity and occupancy rate in intensive care units. Association with mortality. *Intensive Care Medicine* **30**(2) 290–297.

Kc, D., C. Terwiesch. 2007. An empirical analysis of patient flow in the cardiac ICU. Working paper, Wharton School, University of Pennsylvania, Philadelphia.

Kim, S.-C., I. Horowitz, K. K. Young, T. A. Buckley. 1999. Analysis of capacity management of the intensive care unit in a hospital. *Eur. J. Oper. Res.* **115**(1) 36–46.

Kirchhoff, K. T., N. Dahl. 2006. American Association of Critical-care Nurses' national survey of facilities and units providing critical care. *Amer. J. Critical Care* **15**(1) 13–28.

Knaus, W. A., D. P. Wagner, J. E. Zimmerman, E. A. Draper. 1993. Variations in mortality and length of stay in intensive care units. *Ann. Internal Medicine* **118**(10) 753–761.

Lowery, J. C. 1992. Simulation of a hospital's surgical suite and critical care area. *Proc. 1992 Winter Simulation Conf.*, ACM, New York, 1071–1078.

Lowery, J. C. 1993. Multi-hospital validation of critical care simulation model. *Proc. 1993 Winter Simulation Conf.*, ACM, New York, 1207–1215.

Marik, P. E., L. Hedman. 2000. What's in a day? Determining intensive care unit length of stay. *Critical Care Medicine* **28**(6) 2090–2093.

McConnell, K. J., C. F. Richards, M. Daya, S. L. Bernell, C. C. Weathers, R. A. Lowe. 2005. Effect of increased ICU capacity on emergency department length of stay and ambulance diversion. *Ann. Emergency Medicine* **47**(5) 471–478.

McManus, M. L., M. C. Long, A. Cooper, E. Litvak. 2004. Queuing theory accurately models the need for critical care resources. *Anesthesiology* **100**(5) 1271–1276.

Pronovost, P. J., D. M. Needham, H. Waters, C. M. Birkmeyer, J. R. Calinawan, J. D. Birkmeyer, T. Dorman. 2004. Intensive care unit physician staffing: Financial modeling of the Leapfrog standard. *Critical Care Medicine* **32**(6) 1247–1253.

Ryan, S. M. 2004. Capacity expansion for random exponential demand growth with lead times. *Management Sci.* **50**(6) 740–748.

Tu, J. V., C. D. Mazer. 1996. Can clinicians predict ICU length of stay following cardiac surgery? *Canadian J. Anesthesia* **43**(8) 789–794.

Vicente, F. G., F. P. Lomar, C. Mélot, J.-L. Vincent. 2004. Can the experienced ICU physician predict ICU length of stay and outcome better than less experienced colleagues? *Intensive Care Medicine* **30**(4) 655–659.

Weissman, C. 1997. Analyzing intensive care unit length of stay data: Problems and possible solutions. *Critical Care Medicine* **25**(9) 1594–1600.

Yankovic, N., L. V. Green. 2007. A queueing model for nurse staffing. Working paper, Columbia University, New York.