Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

CHARACTERIZING AN EFFECTIVE HOSPITAL ADMISSIONS SCHEDULING AND CONTROL MANAGEMENT SYSTEM: A GENETIC ALGORITHM APPROACH

Jonathan E. Helm
Marcial Lapp
Brendan D. See

Department of Industrial & Operations Engineering
University of Michigan
1205 Beal Avenue, Ann Arbor, MI 48109, USA

ABSTRACT

Proper management of hospital inpatient admissions involves a large number of decisions that have complex and uncertain consequences for hospital resource utilization and patient flow. Further, *inpatient admissions* has a significant impact on the hospital's profitability, access, and quality of care. Making effective decisions to drive high quality, efficient hospital behavior is difficult, if not impossible, without the aid of sophisticated decision support. Hancock and Walter (1983) developed such a management system with documented implementation success, but for each hospital the system parameters are "optimized" manually. We present a framework for valuing instances of this management system via simulation and optimizing the system parameters using a genetic algorithm based search. This approach reduces the manual overhead in designing a hospital management system and enables the creation of Pareto efficiency curves to better inform management of the trade-offs between critical hospital metrics when designing a new control system.

1 INTRODUCTION

Inpatient admissions and bed management is a core value engine of the hospital. Effective management of hospital admissions is critical to the overall cost, quality of care, and patient access. Due to the inherent complexity of the network of resources that encompass hospital care delivery and the dynamic and stochastic nature of patient trajectories within the hospital, effective systems management is difficult without the aid of predictive stochastic models. In the absence of such systems to manage bed and care resources, hospital bed occupancy levels become statistically "out of control," as in the census plot from a partner hospital shown in Figure 1.

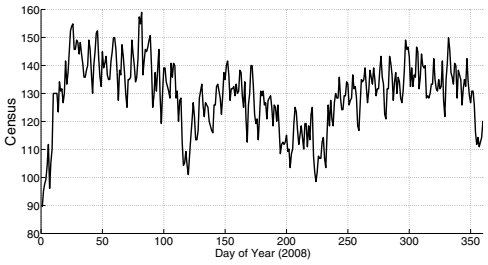


Figure 1: Plot of partner hospital census over the course of one year.

This census variability causes emergency department congestion, elective surgical and medical cancelations, radiology backlogs, strains on nurse and ancillary staff, and intensive care unit (ICU) overcrowding. System-wide congestion caused by high census variability results in compromised quality of care, emergency patient diversions and blockages for lack of beds, increased patient length

of stay (LOS), and significant excess costs (Keehan, Sisko, and Truffer 2007), (Sprivulis et al. 2006), (Harrison, Shafer, and Mackay 2005), and (Proudlove, Gordon, and Boaden 2003).

While a significant amount of research focuses on operating room scheduling or emergency department management, relatively little research examines how the decision to schedule and/or admit a patient affects the downstream hospital resources required to care for that patient over the entire course of their hospitalization. Ignoring the effect of an elective admission on the entire hospital system can contribute significantly to the unstable hospital workloads seen in Figure 1, which leads to the host of problems described above.

Previous research has established effective admission and bed management systems using a variety of modeling techniques. Gallivan and Utley (2005) proposed an integer programming framework for stabilizing hospital workloads. This framework, however, was developed for specialized "elective-only" hospitals in the UK and considers only a single downstream ward or unit for the patients to enter after surgery. This is not the case in most hospitals, in which multiple types of wards serve many different kinds of patients. In addition, there is interaction between wards when patients are transferred due to their dynamically changing condition. In fact, Gupta (2007) states one of the main challenges of modeling surgical admissions is to consider multiple downstream resources over time. In a similar vein, Isken, Ward, and Littig (2010) developed an optimization model for obstetrics, where the decisions include scheduling of c-sections and inductions. Obstetrics generally function somewhat independently of the rest of the hospital, however, and thus constitute a more homogeneous resource. Chow et al. (2008) developed an optimization to reduce congestion in surgical wards, but considers a linear path in which a patient enters a ward and then is discharged with no interaction between wards.

In other research, simulation has been used effectively to account for the complexities of an integrated network of care resources that must serve patients whose needs change dynamically over time. Harper (2002) and Pitt (1997) developed discrete-event simulation frameworks for modeling patient flow and its effect on hospital resources. However, these models are quite general and do not directly constitute decision support for operational management of inpatient admissions.

The Admission Scheduling and Control System developed by (Hancock and Walter 1979) and (Hancock and Walter 1983) and analyzed by (Lowery 1996) is extended in this paper. This system models the hospital as a complex queueing network which encompasses stochastic patient trajectories and the network of resources required to serve patients' dynamically changing needs. The controls on the system are (i) the elective admission schedule, (ii) the census level at which to cancel elective patients, and (iii) the census level at which to call in extra patients off a "callin queue". Hancock and Walter (1983) claim documented savings between $43,000 and $750,000 per year as well as large reductions in surgical cancelations and emergency turnaways based on prior implementations. While this approach to inpatient admissions is effective from a practical standpoint and includes critical features of hospital systems that are omitted in other optimization-based models, it appears that the determination of scheduling parameters is done manually. In this paper, we attempt to develop an optimization framework for automatically generating effective system parameters for such an admission system. To our knowledge, this is the first attempt to add an optimization component to this fully specified control problem of managing admissions to a hospital system, where the system is specified by a *network* of care resources and complete stochastic patient trajectories through the network.

2 METHODS

We model a partner hospital as a proof of concept for our proposed framework for generating an effective admission scheduling and control system. Using this hospital as a case study, we demonstrate that our optimization framework improves upon the existing scheduling and control system. In addition, we demonstrate our framework's usefulness in generating Pareto efficiency curves to guide administrator decision-making.

2.1 Input Modeling

Our simulation uses input data from a mid-size community hospital. To model this hospital, a full year's worth of data is used with identifying patient information removed and replaced by admission numbers. Given that our system is modeling a hospital based on its daily (midnight) census, we only consider patients that stayed in the hospital for at least one night. In 2008, 14,827 patients stayed *at least* one night. Out of these overnight patients, 7,016 were emergency patients while the remaining 7,811 were scheduled patients.

Table 1: Transition probabilities for non-emergency and emergency patients.

	Non-Emergency				Emergency			
To:	–	A	B	C	–	A	B	C
From A	0.846	0.116	0.001	0.037	0.739	0.174	0.008	0.079
From B	0.847	0.004	0.147	0.002	0.699	0.026	0.271	0.004
From C	0.371	0.528	0.069	0.031	0.429	0.543	0.014	0.044

Table 2: Average and standard deviation of the length of stay (in hours) for non-emergency and emergency patients.

	Non-Emergency		Emergency	
Ward	μ	σ	μ	σ
A	27.13	12.87	118.36	131.12
B	23.47	9.68	56.31	65.82
C	21.99	8.20	49.66	123.28

The input data contained the 14,827 patients' movements throughout the hospital. The patients transferred within the hospital 20,462 times, including the initial 'transfer' into the patient's first ward. The transfers within the hospital had been grouped into 23 ward codes by the partner hospital; to avoid unnecessary complexity we aggregate similar wards, where three aggregate wards (which make up 8 of the 23 wards) constitute the majority of transfers (the remaining 15 wards were rarely used by the patients). The first aggregate ward ("Ward A") is a surgical ward. The second aggregate ward ("Ward B") is a medicinal ward. The third aggregate ward ("Ward C") consists of the critical care unit (CCU) and intensive care unit (ICU) of the hospital. This aggregation works well because the statistical properties of patients at the sub-ward level do not differ significantly and the increased sample size for each ward provides better statistical estimates. Since the purpose of our approach is to evaluate system level properties, this loss of granularity has little effect on the system level outcome as noted in Lowery (1996).

2.1.1 Length of Stay and Transition Probabilities

For both emergency and non-emergency patients at each aggregate ward, one-step ward transition probabilities and length of stay parameters are computed. Table 1 gives the one-step transition probabilities for non-emergency and emergency patients. Here, "–" implies a discharged patient. Transitions from a ward back to itself (e.g. a transfer from Ward A to Ward A) may occur for multiple reasons. First, a patient's condition may change, which results in the patient's information (and possibly location within a given ward) being updated. Second, a patient may transfer from a ward in Ward A to another ward which also happens to be in Ward A – thus, even though the patient transfers from one ward to another, our method of aggregating the wards views such movements as internal transfers. We incorporate the transition probabilities into the model as a Markov chain – that is, the probability distribution of a patient's next location solely depends on their current location.

The length of stay data at each aggregate ward is calculated for both types of patients. The mean μ and standard deviation σ at individual wards are weighted appropriately in order to find aggregate versions of these parameters for non-emergency and emergency patients, which are listed in Table 2. We then use these values to find the required parameters for a log-normal distribution ("location" and "scale") for each patient type at each ward. The use of a log-normal distribution to model patients' time spent in a given ward is widely used in the literature (see, e.g., Marshall, Vasilakis, and El-Darzi (2005)). Further, the appropriateness of using a log-normal distribution is evident from multiple probability plots; the plot for one length of stay parameter is given in Figure 2.

2.1.2 Arrival of Emergency Patients

To properly model arrivals, one must consider that emergency patients may (i) arrive at Wards A, B, and C according to a distribution that differs from their intra-hospital transfer rates, and (ii) do

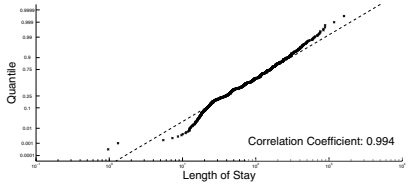


Figure 2: Log-normal probability plot of length of stay parameter.

Table 3: Average number of arrivals, separated by ward and time of day.

	Non-Emergency						Emergency					
Ward:	A		B		C		A		B		C	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
M	8.44	1.38	2.04	1.94	0.35	0.13	1.94	3.17	2.19	2.48	1.15	1.63
Tu	8.45	1.45	6.15	2.19	0.25	0.11	2.03	2.81	1.89	2.53	1.02	1.60
W	7.85	1.32	5.15	1.75	0.25	0.09	1.96	2.87	1.43	2.00	0.92	1.60
Th	9.42	1.18	4.04	1.62	0.17	0.06	1.98	2.50	1.67	2.77	1.23	1.27
F	7.12	0.73	3.19	2.02	0.25	0.08	1.94	3.48	1.67	2.63	1.00	1.44
Sa	0.17	0.12	1.00	0.94	0.02	0.10	1.81	3.48	1.67	2.63	1.00	1.44
Su	1.13	4.87	0.87	1.38	0.12	0.08	1.58	2.17	1.65	2.23	1.12	1.40

not arrive uniformly throughout the week. Consequently, we compute these values as follows. First, we group emergency patients based on the *first* aggregate ward they visit (i.e. Ward A, B, or C). Many patients arrive first to a central triage, and then transfer within the hospital to one of the 23 wards. For each patient, we follow their transfers until they first enter one of the wards aggregated into Ward A, B, or C. For each of the three wards, we determine how many emergency patients are admitted between midnight and 2 p.m. (which we refer to as "AM") and 2 p.m and 11:59 p.m. ("PM") on *each day* of the year. The grouping of emergency patients into AM and PM arrivals is needed to accurately model the system: AM arrivals generally arrive before scheduling decisions are made and thus can be accounted for in those decisions, while PM arrivals enter the hospital after most scheduled patients have been admitted for the day. Further, it is well-known that emergency patients do not arrive uniformly over the course of the day, which leads to increased queueing. The emergency patients in our study generally followed the daily arrival pattern found by previous studies – refer to Draeger (1992) and (See et al. 2009) for empirical distributions.

After determining how many emergency patients arrive at each ward in the AM and PM time blocks for each day of the year, we find the mean and standard deviation of patients arriving at each ward in the AM and PM blocks by day of week. These values are summarized in Table 3. The arrival pattern of non-emergency patients is also of interest in order to evaluate the current system. As one would generally expect, emergency patients do not arrive uniformly over the course of the week, in addition to the heterogeneity over the course of a day. Subsequently, we use the mean number of arrivals (grouped by ward, day of week, and AM/PM) in order to model emergency arrivals using a Poisson distribution.

These calculations – the one-step transition probabilities and length of stay for both emergency and non-emergency patients at Wards A, B, and C, as well as the patients' arrival locations, grouped by day of week and time of day – all serve as inputs to our simulation, which we describe in the next subsection.

2.2 Model Description

Our basic simulation model is similar to that described in the works of Hancock and Walter (1979), Hancock and Walter (1983), and Lowery (1996), which can be referred to for a more detailed de-

scription. As a brief overview, we aggregate our partner hospital's wards into 3 primary wards: Ward A (Surgery), Ward B (Medicine), and Ward C (Critical Care). For each patient type, a flow path is developed based on historical transfer probabilities between wards, as mentioned in §2.1.1. Figure 3 represents the flow paths of different patients in our system. In this system, patients arrive according to their type – Poisson arrivals for emergencies, controlled arrivals for scheduled patients – receive the first segment of treatment and then are either discharged or transferred to another ward for a subsequent treatment segment.

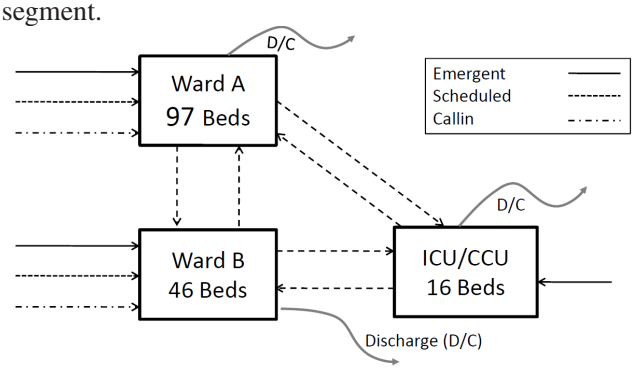


Figure 3: Map of patient flow trajectories.

One complicating feature of hospitals is that many resources within the network are flexible. For example, a surgery patient who exits surgery to find no surgery bed available can occupy a bed in the medicine ward. This is a critical feature of hospital systems that makes it difficult to properly model the hospital system using an optimization framework. In the simulation, alternate routing rules for when a ward is full are encoded in the model based on hospital practice.

Since we intend to use this simulation model for optimization of management system parameters where the simulation will be run thousands of times, one requirement is that a given run of the simulation should complete quickly. Keeping this in mind, the construction of the simulation differs from many discrete-event simulations. Instead of simulating each individual patient arriving to the hospital, we instead simulate and manage groups of patients at once. To do so, we break each day into sequential subcomponents based on the order and timing of the system-level decisions that must be made. Anything that happens in between decision epochs is modeled as a single event. For example, we break the emergencies into AM and PM emergencies. *AM emergencies* are patients that are able to reach a bed before scheduled and callin patients can fill those beds (i.e. before 2 p.m.). *PM emergencies* arrive after the cancel and callin decisions have been made and are thus subject to blockage from the patients admitted that day. We model the arrivals of each group as a single event. This approach highlights the fact that we are modeling system level outcomes and are only concerned with whether or not a patient eventually reaches a bed or is turned away from the hospital in a given day, ignoring the logistics of how that patient reaches the bed.

2.3 Model Control Parameters

We use the same control parameters defined in Hancock and Walter (1983) and Lowery (1996), as our goal is to show that optimization methods can be applied to determine these control parameters rather than designing the scheduling and control system manually. The three parameter types we consider are (i) the *elective admission schedule* by day of week, (ii) the *cancelation level* by day of week – the number of empty beds to leave open for evening emergencies, and (iii) the *callin level* by day of week – the number of empty beds below which we call patients in from the callin queue. Examples of the system controls are shown in Table 4, where each ward that has scheduled patients receives its own set of controls. Setting these 42 control parameters will specify the admission scheduling and control system for this hospital, which could then be implemented as in Hancock and Walter (1983). The goal of this research is to find the controls that will allow the hospital to operate at high efficiency, which means operating at high utilization with limits on the number of cancelations and emergency patient blockages.

The first row of parameters in Table 4 tells the system how many elective patients to admit to each ward by day of week ("schedule"). For example, on Monday 10 patients should be scheduled for

Table 4: Sample choices of the decision variables for Ward A and Ward B.

	Ward A (Surgery)							Ward B (Medicine)						
	Su	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa
Schedule	6	10	8	7	9	9	2	5	9	8	8	9	7	2
Cancel	3	4	4	5	3	2	1	3	5	6	5	4	3	1
Callin	14	12	9	9	9	9	16	15	11	10	10	8	9	14

Ward A and 9 elective patients should be scheduled to enter Ward B. Likewise, the "cancel" decision variable means that if fewer than 4 empty beds remain on a given Monday in Ward A, surgical scheduled patients that would use those beds should be canceled. Finally, the "callin" parameter means that if on a given Wednesday there are only 7 beds full in Ward B after all scheduled patients have been admitted, then the hospital should call in 3 extra patients off the callin queue until 10 beds are filled. This schedule is repeated in the same manner every week, as a repeatable weekly schedule is a prerequisite for the scheduling and control system described in Hancock and Walter (1983), and Lowery (1996).

2.4 Model Validation & Verification

In order to verify and validate the accuracy of our simulation model, we use strategies suggested by Sargent (2005). The verification of the simulation model is done through a series of white-box and black-box testing schemes. In addition to verifying the correct operation of each of the modules illustrated in Figure (3), we also generate patient transition output for each ward for every turn of the simulation clock. That is, using a manual process, we are able to verify that the correct number of patients are flowing through the system on a daily basis, ensuring that our simulation is performing correctly.

We validate our model of the system by comparing it against actual "real-world" hospital operations: given a year's worth of hospital admissions data, we are able to extract the scheduling policy that was used by the hospital and subsequently implement this policy in our model. Comparing the key features of the system (average daily census by day of week, volume of emergency and scheduled patients, and so forth) we find that our simulation closely mimics the actual hospital operations, validating that it indeed functions correctly and produces the correct output.

2.5 Evaluating a Scheduling and Control System

To determine a control parameter set that specifies an effective scheduling and control system we need a mechanism for comparing different systems. To do so, we develop an objective function that embodies the goal of achieving high utilization with limits on the number of cancelations and emergency blockages. The following definitions enable us to formalize an objective function for comparing hospital admission scheduling and control systems.

$X_t(\Theta)$	Random variable denoting the number of cancelations on day $t \in \mathcal{T} = \{Su, M, T, W, R, F, Sa\}$ for a given control parameter set Θ
$Y_t(\Theta)$	Random variable for emergency patient blockages on day t given controls Θ
$Z_t(\Theta)$	Random variable for the number of empty beds at midnight on day t given controls Θ
c	Cancelation cost
b	Blockage cost for emergency patients
h	Cost of an empty bed

Samples from X_t, Y_t, and Z_t are taken daily from the simulation output. While it is possible to further differentiate costs by patient type we do not do so for several reasons. First, the costs are estimates and such differentiation can further complicate the system's functioning. In addition, this system is a high level management system, so it does not tell the hospital *which* patients to cancel, only *how many* to cancel, so the type of patient canceled is subject to doctor and administrator decision-making and cannot be determined within the simulation. Using the following linear cost function we can determine the value, $V(\Theta)$, of a given set of control parameters, Θ, that define a

scheduling and control system:

$$V(\Theta) = \sum_{t \in \mathcal{T}} \left[cX_t(\Theta) + bY_t(\Theta) + hZ_t(\Theta) \right].$$

Note that because we consider scheduling and control parameters that repeat weekly, we are considering a seven day cyclostationary system – if the simulation is run for T days, it generates $T/7$ observations (though certainly not independent) of X_t, Y_t, and Z_t for each day of the week. From these observations, an estimator for $E[V(\Theta)]$ can be obtained. It is this estimator, $\hat{V}(\Theta)$, that is used to compare systems using a genetic algorithm.

While the cost parameters of the objective function cannot be precisely quantified, these parameters can be imputed from hospital management goals. For example, a management goal of "no more than 3 cancelations a month and 2 emergency blockages per month" can be approximately translated into cost parameters that achieve this goal – though this translation is not necessarily unique. By accumulating the objective function over simulation iterations it is possible to estimate a particular scheduling and control system's expected value and thereby possible to compare different scheduling and control systems. This comparison is used in our genetic algorithm to rank members of the population.

2.6 Genetic Algorithm

A genetic algorithm (Davis and Mitchell 1991) is an optimization technique often applied to difficult optimization problems, especially when the state-space of possible solutions is incredibly large and the objective function is non-linear. In our particular problem, the objective is actually the result of a complicated simulation, making optimization approaches rather difficult.

In this particular problem, the input to the genetic algorithm consists of a seven-day hospital scheduling and control policy. In addition, each of the two major wards of the hospital has its own scheduling and control parameters. Thus to characterize the management system for this hospital, our algorithm must determine 42 decision variables that specify the management rules for scheduled patients, callin patients, and cancelations.

To more effectively implement the genetic algorithm, we encode each of the septuplets for each of the wards into a binary bit string. The size of the bit string is determined by the maximum capacity of each ward. For example, the maximum capacity of Ward A is 97 overnight beds, so each Ward A parameter is encoded using a 7-bit binary number.

2.6.1 Implementation

As taken from the general literature, the steps for implementing a typical genetic algorithm are as follows:

1. Generate a population of possible solutions to the problem.
2. Determine the objective function value for each member in the population.
3. Pick two population members, generally ones with highly-ranked objective function values, and combine them to create an offspring ("cross-over").
4. Randomly change genes in the offspring ("mutation").
5. Introduce the offspring into the population and repeat the process.

Using a genetic algorithm to determine the optimal scheduling policy followed the implementation seen in Figure 4.

While a genetic algorithm is by no means guaranteed to provide an optimal solution, it does offer a systematic way to find a better objective function value through an iterative process. For our application, the genetic algorithm needs to be augmented to respect general hospital restrictions. For example, in many hospital systems it is generally the case that patients are not scheduled to arrive at the end of the week because of reduced resource levels on the weekends (Bell and Redelmeier 2001). To respect this constraint, we add to the genetic algorithm restrictions on mutations such that offspring scheduling policies are rejected if they admit too many patients on the weekend. In the case of our partner hospital, we restrict scheduled patients according to the constraints in Table 5. We present the results of the genetic algorithm runs in §3.

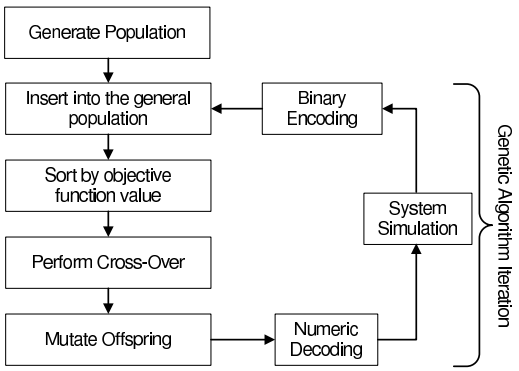


Figure 4: Implementation of the genetic algorithm.

Table 5: Patient scheduling constraints due to reduced weekend resource levels.

	Saturday	Sunday
Ward A	0	≤ 6
Ward B	≤ 2	≤ 2

3 RESULTS

To test our optimization framework, we develop a simulation model using the C++ programming language that reflects the partner hospital described in §2 using the historical data. This simulation model serves as the basis for our genetic algorithm optimization. In this section we first describe the kind of output that was generated and how it was used. Next, we present a comparison of the current system with the system generated from the genetic algorithm solution for a specific cost parameter set. Then we solve the same system under varying parameter sets to generate Pareto efficiency curves that can help guide hospital administrator decision-making rather than forcing the administrator to "choose" a set of costs upon which to optimize. These trade-off curves can be particularly useful for hospital management decision support and at the same time would be quite difficult (if not impossible) to generate manually. This justifies the use of optimization methodologies to solve for effective system parameter sets.

Each hospital simulation is run for 700 days and includes a 34 day warm-up period. This length of time reflects roughly a two-year time frame during which the hospital fluctuations have stabilized. A single replication with 700 days of simulation reduces the error of the objective function to 0.33 at a 97.5% confidence level for a typical objective function whose value ranges between \$800 and \$1,200. Likewise, at a 97.5% confidence level, the error level for the individual objectives (cancelation, turn away, and empty bed costs) is 0.17, 0.35, and 0.04, respectively. These error values indicate that a 700-day horizon is sufficient to attain the steady state of the hospital system.

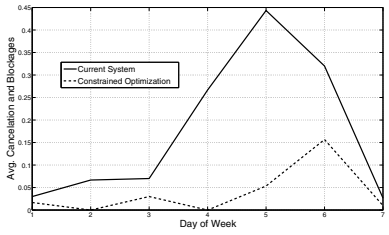
It should be noted that each run of the simulation, including 734 days of operation, completed in 8 seconds on an Intel Core 2 (Duo) E8500 processor running at 3.06Ghz. In addition, the simulation and genetic algorithm execute sequentially and thus do not take advantage of multiple cores/processors. Given the 8 second run-time of a complete simulation, a genetic algorithm run of 3000 iterations completes in just under 7 hours of run-time. It was deemed sufficient to run the algorithm for 3000 iterations at the time of the experiments due to the available machine time and overall processing time required. Further motivation for increased run-time and efficiency are addressed in §4.

3.1 Model Output

3.1.1 Case Study System Comparison

The original hospital suffered from a significant number of cancelations and emergency blockages each month. The goal in implementing an improved inpatient scheduling and control system is to stabilize

the hospital occupancy and enable the hospital to function at the *same utilization* or better with fewer cancelations and blockages. In particular, the goal is to reduce average cancelations and blockages each to fewer than 2 per month. To achieve this goal, different objective function cost parameter sets were tested. Eventually, the parameter set $P = \{h = 1.5, c = 34, b = 45\}$, was found to achieve the stated goal. The results of the genetic algorithm under the cost parameters P are presented in this section.



Mgmt. System	Canceled per mo.	Blocked per mo.	Avg. Util.
Current	7.6	8.1	80%
GA Solution	1.5	1.9	80%

Figure 5: Current system vs. genetic algorithm solution (h = 1.5, c = 34, b = 45).

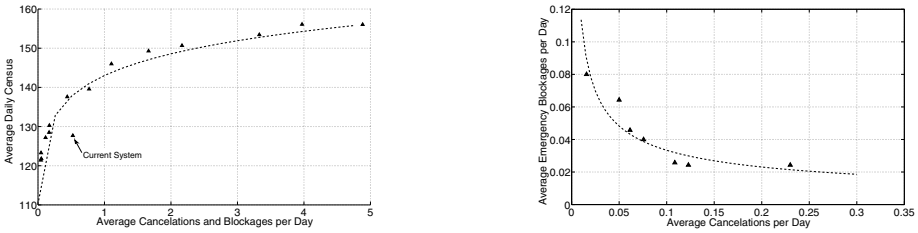
Figure 5 presents a comparison of the simulation results of the current system versus the optimized system. While the cancelations and blockages in the current system spike in the middle of the week (a common occurrence in most hospitals due to uneven scheduling practices), this peak is greatly reduced in the optimized system due to a smoothing of the census across the week. The current system also experiences over 15 cancelations and blockages each month on average, compared with fewer than four per month on average in the optimized system. This is accomplished while still maintaining 80% average utilization (occupancy), which demonstrates the importance of using a management system – improved control systems enable a hospital to significantly improve one set of metrics without negatively impacting a competing metric. In §3.1.2 we will demonstrate how our optimization framework can be used to press the boundaries of competing metrics by generating a Pareto efficiency curve.

3.1.2 Scheduling and Control Pareto Curves

When working with hospitals to design an effective management system it is important to provide the decision makers with as much relevant information as possible. Optimizing to a single objective function as done in §3.1.1 can be informative, but it forces hospital management to identify various costs that cannot be precisely defined. Another approach would be to obtain quantifiable performance goals from the hospital – such as the goal of fewer than two cancelations and two emergency blockages per month with 80% occupancy level or better – and search for a parameter set that achieves these goals. Unfortunately, this process again becomes quite manual and the goals may not even be possible. The approach defined in this section mitigates these difficulties while providing the most information and decision-making flexibility to hospital management by generating Pareto efficiency curves between key system metrics.

By generating Pareto efficiency curves, hospital management can decide what level of service they are willing to accept to achieve a given level of utilization (occupancy) – see Figure 6(a). Creating this curve avoids the difficulties and inaccuracies of estimating specific cost parameters. It also avoids the need to manually search for cost parameters that reflect quantifiable hospital goals by presenting the hospital with possible options and their trade-offs, and allowing management to choose the preferred level for their hospital. Finally, it avoids the situation where management may request an infeasible goal – any point that lies outside the curve is infeasible and need not be considered.

Figure 6 represents the two key trade-offs that are considered in the scheduling and control system analyzed in this paper. A sample of parameter iterates for generating each curve is shown in Table 6. Figure 6(a) represents the trade-off between high utilization (average daily census/occupancy) and congestion (cancelations and blockages). This curve was created by fixing the empty bed cost and iterating over congestion values, where congestion = cancelations + blockages. For each parameter set, the genetic algorithm is used to determine effective scheduling and control system parameters. Notice that the current system lies well within the Pareto efficiency curve, and thus can improve both utilization and congestion simultaneously. This is typically the case in most hospitals that are run without a sophisticated management system.



(a) Plot of utilization vs. congestion for GA solutions

(b) Plot of cancelations vs. blockages at 80% occupancy

Figure 6: Pareto efficiency curves based on GA solutions – obtained by varying cost parameters.

Table 6: Sample of parameter sets used to generate Pareto efficiency curves

	Congestion vs. Utilization					Cancelation vs. Blockage ($\sim$ 80% Util.)				
Cost	Run 1	Run 2	Run 3	Run 4	...	Run 1	Run 2	Run 3	Run 4	...
Empty Bed (h)	1	1	1	1	...	1.4	1.6875	1.875	2.0625	...
Cancelation (c)	0.5	5	10	15	...	30	30	30	30	...
Turnaway (b)	0.5	5	10	15	...	30	37.5	45	52.5	...

Figure 6(b) represents the trade-off between cancelations and emergency patient blockages at a utilization of around 80%, and is generated by maintaining a fixed ratio of congestion cost to utilization cost and varying the ratio of cancelation cost versus the emergency blockage cost. By maintaining a larger safety stock of empty beds through a higher cancelation level, one can reduce the amount of emergency patient blockage. For hospital management to understand the trade-offs and identify the appropriate "safety stock" of empty beds for their hospital, the trade-off curve between cancelations and blockages can be generated for a given level of utilization. In this case, the current system lies so far inside the Pareto curve that it cannot be shown on the graph, signifying an opportunity to improve both cancelations and blockages simultaneously.

By generating these trade-off curves sequentially it is possible to more precisely define the kind of hospital management would like to run. First, one can generate the Pareto curve between utilization and congestion. Once a utilization and congestion level is chosen, it is possible to generate the second efficiency curve between cancelations and blockages at approximately the chosen level of utilization to determine the desired safety stock of empty beds. The average utilization outcome is approximate because one does not choose the utilization level; however one can fix the cost ratios that generated the chosen utilization in the first step to achieve a level close to the desired utilization.

4 DISCUSSION

In this paper, we present a framework that can successfully be used to generate improved scheduling and control policies for hospital systems. Due to the inherent complexity in the overall hospital admissions and scheduling process, simulation is used to determine the effects of such scheduling policies. In combination with a genetic algorithm, we are able to illustrate significant improvements in terms of reduced cancelations and blockages while maintaining high bed utilization for a partner hospital. It should be noted that our simulation models the steady state behavior of a hospital under "normal" operation. In reality, the hospital will deviate from this steady state over the course of the year, however these deviations often occur around holidays and thus are predictable. A complete management system should include a plan for transitioning into and out of holiday periods. This can be done with modifications to our steady state simulation and represents an important area for future work. Additionally, underlying changes in the hospital system dynamics can be tracked via control chart and major changes can be addressed by resimulating to identify the new control parameters.

This optimization framework also facilitates the generation of Pareto efficiency curves as a means of presenting the trade-offs between critical metrics to hospital management. Each point on the efficiency curve represents a different instance of the management system, so hospital administrators can choose the points on the efficiency curves that meet their management objectives. This data point can then be

translated directly into control system parameters. This represents a significant improvement in the system design process, allowing hospital management more flexibility to make the right decision for their hospital. This is made possible because the genetic algorithm is able to automate the generation of efficient management systems, eliminating the need to "optimize" manually.

While genetic algorithms are useful in finding good solutions, they are not guaranteed to return the optimal solution. Due to recent advances in simulation-based optimization approaches, finding the optimal solution even for very large state-space problems has become manageable. One possible extension of our work is to replace the genetic algorithm with an approach such as simulation-based approximate dynamic programming (Si, Barto, Powell, and Wunsch 2004). This strategy has the potential to produce a better solution in less time and could be compared with the genetic algorithm approach presented here.

ACKNOWLEDGMENTS

The authors gratefully appreciate the assistance of the hospital that provided us with the input data. The first author acknowledges the support of the NSF GRFP. The third author gratefully acknowledges the support of NSF-IGERT Grant No. 0654014.

REFERENCES

Bell, C., and D. Redelmeier. 2001. Mortality among patients admitted to hospitals on weekends as compared with weekdays. *New England Journal of Medicine* 345 (9): 663–668.

Chow, V., D. Atkins, W. Huang, M. Puterman, and N. Salehirad. 2008. Reducing surgical ward congestion at the vancouver island health authority through improved surgical scheduling. Technical report, Centre for Operations Excellence - The University of British Columbia.

Davis, L. D., and M. Mitchell. 1991. *Handbook of genetic algorithms*. Van Nostrand Reinhold.

Draeger, M. 1992. An emergency department simulation model used to evaluate alternative nurse staffing and patient population scenarios. In *Proc. of the 1992 Winter Simulation Conference*, ed. J. Swain, D. Goldsman, R. Crain, and J. Wilson, 1592–1600. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gallivan, S., and M. Utley. 2005. Modelling admissions booking of elective in-patients into a treatment centre. *IMA Journal of Management Mathematics* 16 (3): 305–315.

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

Hancock, W., and P. Walter. 1979. The use of computer simulation to develop hospital systems. *SIGSIM Simulation Digest* 10 (4): 28–32.

Hancock, W., and P. Walter. 1983. *The ASCS: Inpatient Admission Scheduling and Control System*. Technical report, Ann Arbor, MI: AUPHA Press.

Harper, P. 2002. A framework for operational modelling of hospital resources. *Health Care Management Science* 5 (3): 165–173.

Harrison, G., A. Shafer, and M. Mackay. 2005. Modelling variability in hospital bed occupancy. *Health Care Management Science* 8 (4): 325–334.

Isken, M., T. Ward, and S. Littig. 2010. An open source project for obstetrical procedure scheduling and occupancy analysis. Under Review.

Keehan, S., A. Sisko, and C. Truffer. 2007. Expenses for hospital inpatient stays: 2004. Technical report, Agency for Healthcare Research and Quality. Statistical Brief.

Lowery, J. 1996. Design of hospital admissions scheduling system using simulation. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. Charnes, D. Morrice, D. Brunner, and J. Swain, 1199–1204. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Marshall, A., C. Vasilakis, and E. El-Darzi. 2005. Length of stay-based patient flow models: recent developments and future directions. *Health Care Management Science* 8 (3): 213–220.

Pitt, M. 1997. A generalised simulation system to support strategic resource planning in healthcare. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradottir, K. Healy, D. Withers, and B. Nelson, 1155–1162. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Proudlove, N., K. Gordon, and R. Boaden. 2003. Can good bed management solve the overcrowding in accident and emergency departments? *British Medical Journal* 20 (2): 149.

Sargent, R. G. 2005. Verification and validation of simulation models. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M. Kuhl, N. Steiger, F. Armstrong, and J. Joines, 130–143. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

See, B., S.-P. Liu, Y.-W. Lu, and Q. Pang. 2009. Staffing a pandemic urgent care facility during an outbreak of pandemic influenza. In *Proc. of the 2009 Winter Simulation Conference*, ed. M. Rossetti, R. Hill, B. Johansson, A. Dunkin, and R. Ingalls, 1996–2007. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Si, J., A. Barto, W. Powell, and D. Wunsch. 2004. *Handbook of learning and approximate dynamic programming*. Wiley-IEEE Press.

Sprivulis, P., J. Da Silva, I. Jacobs, A. Frazer, and G. Jelinek. 2006. The association between hospital overcrowding and mortality among patients admitted via Western Australian emergency departments. *Medical Journal of Australia* 184:208–212.

AUTHOR BIOGRAPHIES

JONATHAN E. HELM is a Ph.D. candidate at the University of Michigan - Ann Arbor in the Industrial and Operations Engineering Department. His research interests lie in patient flow modeling and stochastic optimization as applied to health care delivery processes. He holds a Bachelors degree in Mathematics and Computer Science from Cornell University, a Masters in Operations Research also from Cornell and a Masters in Industrial and Operations Engineering from University of Michigan. His email is <jhelm@umich.edu>.

MARCIAL LAPP is a graduate student in the Industrial and Operations Engineering Department at the University of Michigan. His research interests lie in modeling and solving large-scale optimization problems in the transportation & logistics and health-care industries. He holds a Masters and a Bachelors degree in Computer Science, as well as a Masters of Science Engineering in Industrial and Operations Engineering from the University of Michigan. His email is <mlapp@umich.edu>.

BRENDAN D. SEE is a Ph.D. student at the University of Michigan - Ann Arbor in the Industrial and Operations Engineering Department. He earned a B.S. in Applied Physics and a B.A. in Political Science from the State University of New York (SUNY) at Geneseo and a M.S.E. in Industrial and Operations Engineering from the University of Michigan. His primary research interests lie in procurement, supply chain management, and health care operations management. He can be reached at <bdsee@umich.edu>.